MANUAL

MSX 2+





MSX 2+ FM

Manual MSX 2+ escrito por Ademir Carchano
Manual FM-Music escrito por Flávio Mônaco

Publicado pela:
ACVS Eletrônica Importação e Exportação Ltda.

ÍNDICE

- MSX 2+

- FM-MUSIC

MSX 2+

APRESENTAÇÃO

As informações deste manual permitem que seja usada toda potencialidade do novo MSX BASIC 3.Ø incorporado no cartucho MSX 2+ FM fabricado pela ACVS Eletrônica.

As principais características do computador depois da conexão do cartucho MSX 2+ FM são:

- Texto de até 8Ø colunas (mesmo pela televisão).
- Resolução gráfica máxima de 512 x 424 pontos com 16 cores de 512 combinações (screen 7 modo entrelaçado).
- Resolução máxima de cores de 19.268 (screen 12).
- Memória ROM do BASIC com 64 KBytes (48 KBytes com o MSX BASIC 3.Ø e 16 KBytes com Music-Basic.
- Memória RAM de VÍDEO com 128 KBytes.
- Trabalha com várias páginas gráficas.
- SCREEN Ø a 12.
- SPRITES multicoloridos.
- Pode ter até 8 SPRITES na mesma linha horizontal.
- Relógio-calendário interno mantido por bateria que atualiza hora e data mesmo com o cartucho desconectado.
- Processador de vídeo que executa a maior parte das funções gráficas acelerando o processamento.
- RAMDISK interna com até 32 KBytes.
- Instruções SET PASSWORD, SETTITLE, SETPROMPT,etc.
- Sintetizador com mais 9 canais de som FM.
- 5 sons de instrumentos de percussão (baterias).
- 64 sons de instrumentos musicais pré-programados.
- Saída do sintetizador FM em 2 canais.
- Saídas de vídeo colorido (composto) e de RF (canal 3).

ACVS Eletrônica Importação e Exportação Ltda.

Av. Paulista, 2ØØ1 - 9º andar conj. 912
CEP Ø1311 - São Paulo - SP - Tel. (Ø11) 289-7694

Descrição da sintaxe usada no capítulo MSX 2+

NOME [(X)] (TIPO)

---

1 - O nome do comando, instrução ou função está em letras maiúsculas e deverá ser escrito exatamente como aparece.

2 - Itens entre parênteses ( ) devem ser definidos pelo usuário.

3 - Itens entre colchetes [ ] são opcionais.

4 - Todas as pontuações devem ser escritas como aparecem no formato. Isto se aplica a pontos finais, vírgulas, parênteses, dois pontos, ponto e vírgula, etc.

5 - X, Y e Z representam expressões numéricas.

6 - X$, Y$ e Z$ representam expressões alfanuméricas.

(TIPO) - Indica se é um COMANDO, uma INSTRUÇÃO ou FUNÇÃO. A diferença entre os tipos é a seguinte:

- COMANDO: é a ordem que, depois de executada, coloca o computador no modo de comando direto via teclado. Por exemplo: LIST.

- INSTRUÇÃO: é a ordem que pode ser dada via teclado ou dentro de um programa. Por exemplo PRINT, GOTO, GOSUB, etc.

- FUNÇÃO: só é válida acompanhada por uma INSTRUÇÃO.

Por exemplo: PRINT BASE(Ø)

SCREEN [X [,Y [,Z [,XX [,YY [,ZZ]]]]]] Instrução

Define o modo de tela que será usado, o tamanho dos SPRITES, o chaveamento do CLICK do teclado, a velocidade (baud rate) do gravador cassete e o tipo de impressora utilizada.

X é o modo de tela que será usado. Podendo ser:

- SCREEN Ø - modo texto 1. Com até 8Ø caracteres por linha num máximo de 26.5 linhas. Pode assumir até duas cores, uma para o caractere e outra para a tela de fundo. O número de caracteres é definido pela instrução WIDTH. As cores são definidas pelo comando COLOR. Cada caractere é montado numa matriz com 6 pontos na horizontal e 8 na vertical.
- SCREEN 1 - modo texto 2. Com até 32 caracteres por linha num máximo de 26.5 linhas. Trabalha com até 16 cores de 512 combinações e pode ter uma cor para os caracteres, uma para o fundo e outra para a borda da tela. A matriz de caracteres é formada por um quadrado de 8 por 8 pontos. Usa os SPRITES no modo 1 (ver item tipos de SPRITE).
- SCREEN 2 - modo gráfico 1. Com 256 pontos na horizontal por 192 pontos na vertical e 16 cores de 512 combinações. Podem ser usadas somente duas cores a cada 8 pontos na horizontal, na vertical não há esta limitação. Usa os SPRITES no modo 1 (ver item tipos de SPRITE).
- SCREEN 3 - modo gráfico 2 - 64 pontos na horizontal por 48 na vertical. Cada ponto é formado por um quarto do tamanho normal de um caractere na SCREEN 1. Usa 16 cores de 512 combinações. Usa os SPRITES no modo 1 (ver item tipos de SPRITE).
- SCREEN 4 - modo gráfico 1. Idêntica à SCREEN 2 só que usa os SPRITEs no modo 2 (ver item Tipos de Sprite).
- SCREEN 5 - modo gráfico com bit mapeado - resolução de 256 pontos na horizontal e 212 na vertical. Trabalha com 16 cores ponto a ponto de uma paleta de 512 combinações. Possui 4 páginas gráficas e usa os sprites no modo 2.

- SCREEN 6 - modo gráfico com bit mapeado de alta resolução - 512 pontos na horizontal por 212 na vertical. Trabalha com 4 cores ponto a ponto de uma paleta de 512 combinações. Quatro páginas gráficas e sprites no modo 2.

- SCREEN 7 - modo gráfico com bit mapeado de alta resolução - 512 pontos na horizontal por 212 na vertical. Trabalha com 16 cores ponto a ponto de uma paleta de 512 combinações. Duas páginas gráficas e sprites no modo 2.

- SCREEN 8 - modo gráfico com bit mapeado - resolução de 256 pontos na horizontal por 212 na vertical e 256 cores simultâneas ponto a ponto. Duas páginas gráficas e sprites no modo 2.

- SCREEN 9 - não existe nos MSX 2+, mesmo nos importados.

- SCREEN 1Ø - modo gráfico 1 com cores no sistema YJK - 256 pontos na horizontal e 212 na vertical. Trabalha com até 12.499 cores do sistema YJK mais as 16 cores básicas (Ø a 15) da paleta de 512 combinações. Neste tipo de tela os comandos gráficos trabalham somente com cores usando valores entre Ø e 15. Portanto, pode-se traçar linhas sobre uma tela feita no sistema YJK sem ocorrer borrões na tela de fundo. Tem duas páginas gráficas e usa os sprites no modo 2.

- SCREEN 11 - modo gráfico 1 com cores no sistema YJK - 256 pontos na horizontal por 212 na vertical. Trabalha com até 12.499 cores. O valor das cores nos comandos gráficos pode variar entre Ø e 255. Dependendo do valor da cor e do tipo de comando gráfico podem ocorrer borrões entre as cores no limite de quatro em quatro pontos. Tem duas páginas gráficas e usa sprites no modo 2.

- SCREEN 12 - modo gráfico 2 com cores no sistema YJK - 256 pontos no horizontal por 212 na vertical. Trabalha com até 19.268 cores de 131Ø72 combinações. O limite para troca abrupta de cores é de quatro em quatro pontos na horizontal. Tem duas páginas gráficas e usa os prites no modo 2.

Tipos de SPRITES: nos MSX 1 os SPRITES pode ter somente uma cor e no máximo 4 SPRITES podem estar na mesma linha horizontal. Nos MSX 2+ os SPRITES podem ser

multicoloridos (cada linha da figura pode ter uma cor) e até 8 (oito) SPRITES podem ser colocados na mesma linha horizontal. Também podem ser formadas figuras mais complexas sobrepondo-se vários SPRITES, pois existe a posibilidade de se desativar o sensor de coincidência de SPRITES (bit que é ativado sempre que dois ou mais SPRITES se tocam).

A resolução vertical pode ser dobrada para 424 usando-se o modo entrelaçado de exibição. (ver item ZZ).

Y determina o tamanho dos SPRITES:

Ø- 8x8 mag. 1 (8x8)   2- 16x16 mag. 1 (16x16)
1- 8x8 mag. 2 (16x16) 3- 16x16 mag. 2 (32x32)

Z ativa ou desativa o CLICK do teclado:

Ø - desativa o CLICK   1 - ativa CLICK

XX define a velocidade de transmissão dos dados para o gravador cassete (baud rate):

1 - 12ØØ baud   2 - 24ØØ baud

YY define o tipo de impressora usada:

Ø - MSX   1 - não MSX

ZZ define o modo de exibição da tela. Podendo ser:

Ø - Normal
1 - Entrelaçado
2 - Não entrelaçado, alternando páginas impares e pares
3 - Entrelaçado, alternando páginas impares e pares.

Nos modos de exibição 2 e 3, o número da página de exibição deve ser impar (ver instrução SETPAGE).

Ex:

```
1Ø COLOR 15,Ø,Ø
2Ø FOR I= 2 TO 8 : SCREEN I
3Ø FOR N=Ø TO 255:LINE (Ø,Ø)-(N,211),C
4Ø IF I=8 THEN C=N ELSE C=C+1:IFC=16THENC=1
5Ø NEXT
6Ø FOR T=Ø TO 1ØØØ:NEXT
7Ø NEXT
```

```
1Ø SCREEN 7,,,,,3
2Ø LINE(8Ø,8Ø)-(4ØØ,2ØØ),4,BF
3Ø SETPAGE1,1:CLS
4Ø LINE(1ØØ,1ØØ)-(38Ø,18Ø),4,BF
5Ø GOTO5Ø
```

COLOR [X] [,Y] [,Z] — Instrução

Define as cores que serão utilizadas na tela nos modos texto e gráfico.

X define a cor do caractere, Y define a cor do fundo da tela e Z define a cor da borda da tela. X, Y e Z devem ser números inteiros.

Nas SCREENs Ø, 1, 2, 3, 4, 5 e 7, os números X, Y e Z devem estar compreendidos entre Ø e 15.

Nas SCREENs 5 e 7 cada byte guarda a cor de dois pontos em sequência.

A cor correspondente a cada número na paleta de cores é mostrada na tabela a seguir:

Nº na paleta	Nome da cor	Intensidade: Verm.	Verde	Azul
Ø	transparente	Ø	Ø	Ø
1	preta	Ø	Ø	Ø
2	verde média	1	6	1
3	verde clara	3	7	3
4	azul escura	1	1	7
5	azul clara	2	3	7
6	vermelha escura	5	1	1
7	ciano	2	6	7
8	vermelha média	7	1	1
9	vermelha clara	7	3	3
1Ø	amarela escura	6	6	1
11	amarela clara	6	6	4
12	verde escura	1	4	1
13	magenta	6	2	5
14	cinza	5	5	5
15	branca	7	7	7

A intensidade do vermelho, verde e azul na paleta de cores pode ser alterada com a instrução COLOR=.

Na SCREEN 6, X, Y e Z devem estar entre Ø e 3. Neste modo gráfico, se a cor da borda da tela estiver entre Ø e 3, será respeitada a tabela da paleta de cores. Porém, se a cor estiver entre 16 e 31, será feita uma combinação das cores conforme a tabela a seguir:

Nº da cor da borda	Combinação entre as cores:
16	Ø e Ø
17	Ø e 1
18	Ø e 2
19	Ø e 3
2Ø	1 e Ø
21	1 e 1
22	1 e 2
23	1 e 3
24	2 e Ø
25	2 e 1
26	2 e 2
27	2 e 3
28	3 e Ø
29	3 e 1
3Ø	3 e 2
31	3 e 3

Na SCREEN 8, X, Y e Z devem estar entre Ø e 255. Neste modo gráfico cada byte corresponde a cor de um ponto. Para saber o número da cor de acordo com a intensidade de vermelho, verde e azul, use a equação abaixo:

G * 32 + R * 4 + B

onde G, R e B são respectivamente as intensidades de verde, vermelho e azul.

O verde e o vermelho podem variar entre Ø e 7. O azul só pode variar entre Ø e 3.

A formação do byte da cor na SCREEN 8 é a seguinte:

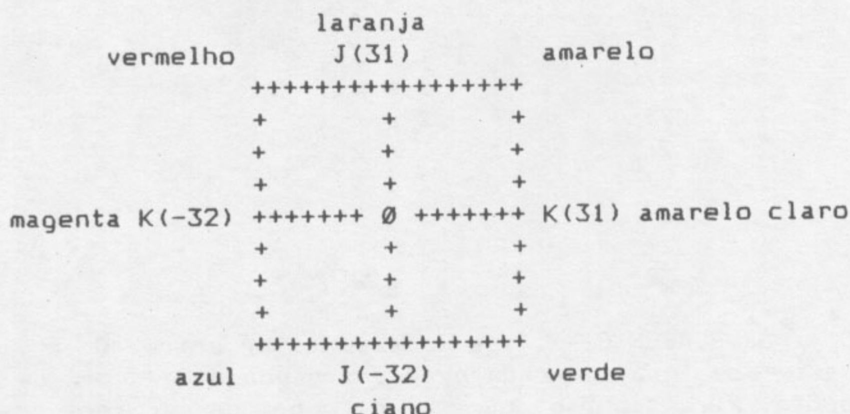
bit -	7	6	5	4	3	2	1	Ø
cor -	G	G	G	R	R	R	B	B

Nas SCREEN 1Ø, 11 e 12 é usado o sistema YJK para mostrar as cores. Neste modo especial, 4 (quatro) bytes são usados para compor as cores de quatro pontos em sequência na mesma linha horizontal:

```
   BIT -  7   6   5   4   3   2   1   Ø          Pontos:
                                                1  2  3  4
Byte 1 -  Y1  Y1  Y1  Y1  Y1  KL  KL  KL  --- o  o  o  o
Byte 2 -  Y2  Y2  Y2  Y2  Y2  KH  KH  KH  -------+  +  +
Byte 3 -  Y3  Y3  Y3  Y3  Y3  JL  JL  JL  ----------+  +
Byte 4 -  Y4  Y4  Y4  Y4  Y4  JH  JH  JH  -------------+
```

Os (Y) são a intensidade da cor (brilho) e podem variar de Ø a 31. Cada valor de Y corresponde a 8 (oito) níveis na escala de intensidade (Ø=Ø, 1=8, 2=16,..., 31=248).

Os (J e K) são os vetores que determinam qual cor será usada. Podem variar de -32 a 31 (6 bits). O gráfico a seguir mostra como é feita a seleção da cor:



A conversão de um sistema para outro é feita com as seguintes equações:

De YJK para RGB

$R = Y + J$

$G = Y + K$

$B = 5/4\ Y - 1/2\ J - 1/4\ K$

De RGB para YJK

$Y = B/2 + R/4 + G/8$

$J = R - Y$

$K = G - Y$

Nas SCREENs 1Ø e 11, os bytes das cores têm o seguinte formato:

BIT -	7	6	5	4	3	2	1	Ø
PONTO 1 -	Y1	Y1	Y1	Y1	A	KL	KL	KL
PONTO 2 -	Y2	Y2	Y2	Y2	A	KH	KH	KH
PONTO 3 -	Y3	Y3	Y3	Y3	A	JL	JL	JL
PONTO 4 -	Y4	Y4	Y4	Y4	A	JH	JH	JH

O Y de cada ponto pode variar de Ø a 15 (Ø a 24Ø na intensidade). K e J permanecem como na SCREEN 12 variando de -32 a 31.

SCREEN 1Ø - as cores são formadas no sistema YJK ou com a paleta normal de 16 cores, dependendo do bit 3 (A). Se for Ø, as cores são mostradas no sistema YJK, considerando o Y de 4 bits (Ø a 24Ø de intensidade), obtendo-se assim as 12.499 variações possíveis. Se o bit

3 (A) for 1, as cores são as da paleta de cores conforme os bits Y (Ø a 15). Neste caso, os bits J e K são ignorados. As instruções gráficas nesta SCREEN podem usar valores para cor entre Ø e 15.

SCREEN 11 - as cores são formadas como na SCREEN 1Ø só que as instruções gráficas trabalham com valores para a cor entre Ø e 255.

Quando não for dado nenhum valor a X, Y ou Z, a instrução COLOR utilizará os últimos valores aplicados às cores dos caracteres, fundo e borda da tela.

Ex1:

```
1Ø SCREEN7
2Ø FORX=ØTO511
3Ø LINE(255,Ø)-(X,211),C
4Ø C=C+1:IFC=16THENC=Ø
5Ø NEXT
6Ø GOTO6Ø
```

Ex2:

```
1Ø SCREEN8
2Ø FORX=ØTO255
3Ø LINE(Ø,Ø)-(X,211),X
4Ø LINE(255,211)-(255-X,Ø),X
5Ø NEXT
6Ø GOTO6Ø
```

Ex3:

```
1ØØ SCREEN12
11Ø Y=93
12Ø FOR JH=Ø TO 3
13Ø FOR JL=Ø TO 7
14Ø FOR KH=Ø TO 3
15Ø FOR KL=Ø TO 7
16Ø LINE (X,Y)-(X,Y+2),KL,BF
17Ø LINE (X+1,Y)-(X+1,Y+2),KH,BF
18Ø LINE (X+2,Y)-(X+2,Y+2),JL,BF
19Ø LINE (X+3,Y)-(X+3,Y+2),JH,BF
2ØØ X=X+4 : IFX=128 THEN X=Ø : Y=Y-3
21Ø NEXT KL,KH,JL,JH
22Ø COPY (Ø,Ø)-(127,95) TO (128,Ø)
23Ø COPY (Ø,Ø)-(255,95) TO (Ø,96)
24Ø FOR X=Ø TO 127 STEP 4
```

```
250 LINE (X+1,Ø)-(X+1,191),4,,OR
260 NEXT X
270 FOR X=Ø TO 255 STEP 4
280 LINE (X+3,96)-(X+3,191),4,,OR
290 NEXT X
300 FOR N=Ø TO 31
310 LINE (Ø,Ø)-(255,191),7,BF,AND
320 LINE (Ø,Ø)-(255,191),CC,BF,OR
330 FOR T=Ø TO 3ØØ : NEXT T
340 CC=CC+8
350 NEXT N
360 IF INKEY$="" THEN 3ØØ
```

COLOR=(X, R, G, B) — Instrução

Altera as intensidades do vermelho, verde e azul na paleta de cores.

X é o número da cor na paleta de cores e deve estar entre Ø e 15.

R é a intensidade do vermelho entre Ø e 7.

G é a intensidade do verde entre Ø e 7.

B é a intensidade do azul e deve estar entre Ø e 7.

A SCREEN 8 não usa a paleta de cores.

Ex:

```
100 SCREEN5
110 OPEN"GRP:"AS#1
120 FORN=ØTO7
130 PRESET(N*32+6,4Ø):PRINT#1,"COR"
140 PRESET(N*32+6,5Ø)
150 PRINT#1,N+1
160 NEXT
170 FORX=ØTO7
180 LINE(X*32,6Ø)-(X*32+32,14Ø),X+1,BF
190 LINE(X*32+32,141)-(X*32+64,211),15-X,BF
200 LINE(X*32,35)-(X*32+32,211),1Ø,B
210 NEXT
220 PRESET(Ø,1Ø)
230 FORN=ØTO7
240 PRINT#1," RGB";
250 NEXT
```

```
260 FORC=ØTO7
270 COLOR=(C+1,R,G,B)
280 R$=RIGHT$(STR$(R),1)
290 G$=RIGHT$(STR$(G),1)
300 B$=RIGHT$(STR$(B),1)
310 PRESET(C*32+8,20):PRINT#1,R$;G$;B$
320 B=B+1:IFB=8THENB=Ø:G=G+1:IFG=8THENG=Ø:R=R+1:IFR=8
THENR=Ø
330 NEXT
340 GOTO260
```

COLOR=NEW — Instrução

Volta as intensidades de vermelho, verde e azul da paleta de cores aos seus valores originais.

Os valores originais de vermelho, verde e azul são aqueles mostrados na tabela junto à instrução COLOR. Ex:

```
10 SCREEN 5 : COLOR 15,Ø,Ø
20 OPEN "GRP:" AS #1
30 FOR C=2 TO 15
40 COLOR C
50 PRINT#1,"ALTERAÇÃO DAS CORES DA PALETA"
60 NEXT C
70 FOR N=2 TO 15
80 R=INT(RND(1)*7)
90 G=INT(RND(1)*7)
100 B=INT(RND(1)*7)
110 COLOR=(N,R,G,B)
120 NEXT N
130 FOR T=Ø TO 1000 : NEXT T : BEEP
140 COLOR = NEW
150 IF INKEY$="" THEN 15Ø
```

COLOR=RESTORE — Instrução

Transfere da memória de vídeo para a paleta de cores os valores das intensidades do vermelho, verde e azul.

Esta instrução é usada quando os valores das intensidades do vermelho, verde e azul são carregados para a memória de vídeo com a instrução BLOAD"nome do arquivo",S.

A tabela da paleta de cores pode ser encontrada nos seguintes endereços na memória de vídeo:

SCREEN Ø (4Ø colunas)	&HØ4ØØ - &HØ42Ø
SCREEN Ø (8Ø colunas)	&HØFØØ - &HØF2Ø
SCREEN 1	&H2Ø2Ø - &H2Ø4Ø
SCREEN 2	&H2Ø2Ø - &H2Ø4Ø
SCREEN 3	&H2Ø2Ø - &H2Ø4Ø
SCREEN 4	&H2Ø2Ø - &H2Ø4Ø
SCREEN 5	&H768Ø - &H76AØ
SCREEN 6	&H768Ø - &H76AØ
SCREEN 7	&HFA8Ø - &HFAAØ
SCREEN 8	&HFA8Ø - &HFAAØ

Nas SCREENs 5, 6, 7 e 8 pode-se trabalhar com mais de uma página de vídeo (ver instrução SET PAGE). Para descobrir o endereço da tabela da paleta de cores, use o seguinte cálculo:

SCREEN 5 e 6 --> número da página * &HØ8ØØØ + &H768Ø
SCREEN 7 e 8 --> número da página * &H1ØØØØ + &HFA8Ø

Ex:

```
1Ø SCREEN Ø : WIDTH 4Ø : COLOR 15,Ø,Ø
2Ø LOCATE 1Ø,1Ø : PRINT "COR 15 NORMAL"
3Ø FOR T=Ø TO 5ØØ : NEXT : BEEP
4Ø COLOR= (15,3,2,1)
5Ø LOCATE 1Ø,1Ø : PRINT "COR 15 ALTERADA"
6Ø BSAVE "TESTECOR.PIC",&HØ4ØØ,&HØ42Ø,S
7Ø COLOR=NEW : BLOAD "TESTECOR.PIC",S
8Ø FOR T=Ø TO 5ØØ : NEXT : BEEP
9Ø COLOR=RESTORE
```

COLORSPRITE (X) = Y — Instrução

Define a cor e algumas características de um SPRITE.

Esta instrução só funciona nos modos gráficos de SCREEN 4 em diante.

X é o número do SPRITE e deve ser igual ao número

na instrução SPRITE$ que define as características do SPRITE.

Y deve ser um número inteiro compreendido entre Ø e 111. E tem o seguinte significado:

- Entre Ø e 15, são os números das cores da paleta de cores.

- Somando 32 ao número da paleta de cores, além da cor escolhida, o SPRITE não ativa mais o sensor de coincidência de SPRITES. Ou seja, se este SPRITE colidir com outro na tela a instrução ON SPRITE GOSUB não será executada.

- Somando 64 ao número da paleta de cores, além da cor escolhida, o SPRITE não ativa mais o sensor de coincidência de SPRITE e faz uma operação lógica OR com as cores dos SPRITES que estão sobrepostos.

As cores dos SPRITES podem ser determinadas pelas instruções: PUTSPRITE, COLOR SPRITE ou COLOR SPRITE$. A instrução que for executada por último é que determinará a cor do SPRITE. Quando for usar as instruções COLOR SPRITE ou COLOR SPRITE$, não defina o número da paleta de cores na instrução PUT SPRITE.

Para detalhes da intrução PUT SPRITE, recorra ao manual do MSX1.

Ex:

```
100 SCREEN5,1:COLOR15,Ø,Ø
110 FORX=ØTO15:LINE(X*16,4Ø)-STEP(16,12Ø),15-X,BF:NEXT
120 B$=""
130 FORI=1TO8:READA$:B$=B$+CHR$(VAL("&b"+A$)):NEXT
140 SPRITE$(Ø)=B$
150 SPRITE$(1)=B$
160 COLORSPRITE(1)=33 'SOMBRA
170COLORSPRITE$(Ø)=CHR$(12)+CHR$(1Ø)+CHR$(5)+CHR$(8)+CHR
$(3)+CHR$(3)
180 FORI=ØTO212
190 PUTSPRITEØ,(I,I),,Ø
200 PUTSPRITE1,(I+8,I+8),,1
210 NEXT
220 GOTO22Ø
230 DATA 1ØØ11ØØ1
240 DATA ØØ1111ØØ
250 DATA Ø111111Ø
260 DATA 11111111
```

```
270 DATA 11111111
280 DATA Ø111111Ø
290 DATA Ø1ØØØØ1Ø
300 DATA 1ØØØØØØ1
```

COLORSPRITE$ (X) = Y$ — Instrução

Define a cor das linhas de um SPRITE.

Esta instrução só funciona nos modos gráficos de SCREEN 4 em diante.

X é o número do SPRITE e deve ser igual ao número na instrução SPRITE$ que define as características do SPRITE.

Y$ deve ser uma expressão com 1 até 16 caracteres, cada um correspondendo a uma linha do SPRITE. O código dos caracteres, de acordo com o seu valor, tem o seguinte significado:

Entre Ø e 15, são os números das cores da paleta de cores.

Somando 32 ao número da paleta de cores, além da cor escolhida, a linha do SPRITE não ativa mais o sensor de coincidência de SPRITEs. Ou seja, se a linha do SPRITE colidir com outra na tela, a instrução ON SPRITE GOSUB não será executada.

Somando 64 ao número da paleta de cores, além da cor escolhida, a linha do SPRITE não ativa mais o sensor de coincidência de SPRITEs e faz uma operação lógica OR com as cores dos SPRITEs que estão sobrepostas.

Ao se somar 128 ao número da paleta de cores, a linha do SPRITE se deslocará 32 pontos para a esquerda.

As cores dos SPRITES podem ser determinadas pelas instruções: PUTSPRITE, COLOR SPRITE ou COLOR SPRITE$. A instrução que for executada por último é que determinará a cor do SPRITE. Quando for usar as instruções COLOR SPRITE ou COLOR SPRITE$, não defina o número da paleta de cores na instrução PUT SPRITE.

Para detalhes da instrução PUT SPRITE, recorra ao manual do MSX 1.Ø.

Exemplo: ver exemplo na instrução COLOR SPRITE.

COPY origem TO destino — Instrução

```
COPY(X1,Y1)-[STEP](X2,Y2)[,XX] TO (X3,Y3)[,YY] [,op lóg]
COPY (X1,Y1) - [STEP] (X2,Y2) [,XX] TO matriz
COPY (X1,Y1)-[STEP](X2,Y2)[,XX] TO "perif: nome arquivo"
COPY matriz[,sentido] TO (X3,Y3)[,YY] [,operação lógica]
COPY matriz TO "periférico: nome arquivo"
COPY"perif:nome arq."[,sentido]TO (X3,Y3)[,YY] [,op lóg]
COPY "perif: nome arquivo1" TO "perif: nome arquivo2"
```

Copia dados de uma tela gráfica para outro ponto da tela, para outra página de tela gráfica, para uma matriz ou para um arquivo em disquete. Traz dados de uma matriz ou que estejam gravados em disquetes para a tela gráfica. Copia um arquivo de um disquete para outro, ou faz cópia do arquivo no mesmo disquete com outro nome.

Esta instrução só é válida nas SCREENs 5, 6 ,7 e 8. X1 e Y1 são as coordenadas na tela gráfica do início do bloco que será copiado.

X2 e Y2 são as coordenadas na tela gráfica do fim do bloco que será copiado. Se for usada a palavra STEP antes dos números (X2,Y2), estes números passam a ser a quantidade de pontos nos eixos X e Y que serão somados às coordenadas X1 e Y1, respectivamente. Os resultados das somas serão as coordenadas finais do bloco.

X3 e Y3 são as coordenadas de destino, ou seja, ponto na tela gráfica para onde os dados começarão a ser transferidos.

X1, X2 e X3 são números inteiros e podem variar entre Ø e 255 nas SCREENs 5 e 8, e entre Ø e 511 nas SCREENs 6 e 7.

Y1, Y2 e Y3 podem variar entre Ø e 211 nas SCREENs 5,6,7 ou 8.

Nas SCREENs 5, 6, 7 e 8 você pode trabalhar com mais de uma página de vídeo (ver instrução SET PAGE). Nestes casos:

XX e YY representam o número da página de origem e o número da página de destino respectivamente. Se XX e YY não forem determinados, será assumida a página que está sendo usada.

[Sentido]: determina o sentido em que os dados serão transferidos. Podem variar entre Ø e 3 e têm o seguinte significado:

Ø - da esquerda superior para a direita inferior.
1 - da direita superior para a esquerda inferior.
2 - da esquerda inferior para a direita superior.
3 - da direita inferior para a esquerda superior.

Matriz: é o nome da variável (matriz numérica) que deve ter tamanho suficiente para receber os dados da tela gráfica. O tamanho da matriz pode ser calculado com a seguinte equação:

INT ((pixel * (ABS(X2-X1)+1) * (ABS(Y2-Y1)+1) +7)/8)+4

Pixel nas SCREENs 5, 7 e 8 vale quatro e na SCREEN 6 vale dois.

[Operação]: é uma operação lógica que pode ser feita entre as cores de origem e as cores de destino da área copiada. O resultado da operação é visto na região de destino. As operações lógicas podem ser:

OR - C2 = C1 + C2
XOR - C2 = NOT(C1) * C2 + C1 * NOT(C2)
AND - C2 = C1 * C2
PSET - C2 = C1
PRESET - C2 = NOT(C1)

TOR, TXOR, TAND, TPSET e TPRESET, executam as mesmas operações lógicas descritas acima com a diferença de não afetarem a cor transparente (Ø).

C1 é o número da cor de origem.
C2 é o número da cor de destino.
NOT(cor) significa a inversão bit a bit do número da cor.

Periférico: (Perif) é a letra que indica qual controlador de disquete será usado, podendo ser: A, B, C, D, E ou F dependendo do número de interfaces de drives ligadas ao computador.

Nome arquivo1 e nome arquivo2 são, respectivamente, os nomes dos arquivos de origem e de destino que são gravados no disquete.

Ex:

```
100 SCREEN7,Ø:COLOR15,Ø,Ø
110 X1=Ø:Y1=Ø:X2=64:Y2=32
120 TM=INT((4*(ABS(X2-X1)+1)*(ABS(Y2-Y1)+1)+7)/8)+4
130 DIM A(TM)
```

```
140 FORX=1TO7
150 CIRCLE(16,16),X*2-1,X:CIRCLE(16,16),X*2,X
160 CIRCLE(48,16),X*2-1,8+X:CIRCLE(48,16),X*2,8+X
170 NEXT
180 FORC=1TO7
190 COLOR=(C,Ø,Ø,7-C/2):COLOR=(C+8,7-C/2,Ø,Ø)
200 NEXT
210 COPY(Ø,Ø)-STEP(64,32)TOA
220 FORN=1TO100
230 X=RND(N)*256:Y=RND(N)*200
240 COPYA,ØTO(X,Y),,TPSET
250 NEXT
260 COPY(Ø,Ø)-(350,211)TO(200,Ø),,TPSET
270 GOTO270
```

SETPAGE X,Y — Instrução

Seleciona a página da tela que será exibida e a página que será ativa (na qual os desenhos serão feitos).

Esta instrução só é válida nas SCREENs 5 a 12

X determina a página que será exibida e deve ser um número inteiro entre Ø e 3, dependendo da SCREEN que está sendo utilizada.

Y determina a página que será ativa e deve ser um número inteiro entre Ø e 3, dependendo da SCREEN que está sendo utilizada.

As páginas de cada SCREEN são:

SCREEN 5 - 4 páginas (Ø a 3)
SCREEN 6 - 4 páginas (Ø a 3)
SCREEN 7 - 2 páginas (Ø e 1)
SCREEN 8 - 2 páginas (Ø e 1)
SCREEN 10 - 2 páginas (Ø e 1)
SCREEN 11 - 2 páginas (Ø e 1)
SCREEN 12 - 2 páginas (Ø e 1)

Ex:

```
10 SCREEN 5: COLOR 15,Ø,Ø
20 LINE (20,20)-(220,200),4,BF
30 SETPAGE Ø,1:CLS
40 LINE (40,40)-(200,180),8,BF
```

```
50 SETPAGE Ø,2:CLS
60 LINE (60,60)-(180,160),12,BF
70 SETPAGE Ø,3:CLS
80 CIRCLE (127,105),80,4
90 PAINT (127,105),10,4
100 FOR P=Ø TO 3
110 SETPAGE P,P
120 FOR T=Ø TO 300 :NEXT T:BEEP
130 NEXT P
140 GOTO 100
```

SETSCROLL [X] [,Y] [,Z] [,XX] — Instrução

Determina uma coordenada que será usada como ponto inicial, a partir do qual a tela será mostrada no vídeo.

X é um número entre Ø e 511 e define a coordenada horizontal.

Y é um número entre Ø e 255 e define a coordenada vertical.

Z determina se será feito SCROLL nas 8 colunas da esquerda. Podendo ser:

Ø - as 8 colunas da esquerda da tela são mostradas.

1 - as 8 colunas da esquerda da tela não aparecem.

XX determina se haverá troca de páginas de vídeo ou não. Podendo ser:

Ø - somente uma página. Neste caso, quando a tela desaparece em uma das laterais do vídeo, a mesma tela reaparece no lado oposto.

1 - duas páginas. Neste caso, quando a primeira página desaparece em uma das laterais do vídeo, a segunda página aparece no lado oposto.

Para movimentar duas telas, a página ativa deve ser impar. Para isso, use a instrução SETPAGE.

Ex:

```
10 SCREEN 8,Ø
20 COPY (Ø,Ø)-(255,80) TO (Ø,211)
30 FOR X=Ø TO 255
40 LINE (255,Ø)-(X,211),X
50 NEXT
60 SETPAGE 1,1:CLS
```

```
70 COPY (Ø,Ø)-(255,8Ø) TO (Ø,211)
8Ø FOR X=Ø TO 255
9Ø LINE (Ø,Ø)-(X,211),X
1ØØ NEXT
11Ø FOR X=ØTO511
12Ø SET SCROLL X,Ø,1,1
13Ø NEXT
14Ø FOR X=Ø TO 511
15Ø SETSCROLL X,X/2,1,1
16Ø NEXT
17Ø GOTO 11Ø
```

WIDTH X — Instrução

Determina o número de caracteres que será usado por linha de texto.

X é um número inteiro que pode variar de 1 a 8Ø na SCREEN Ø e de 1 a 32 na SCREEN 1.

Na SCREEN Ø se X for menor ou igual a 4Ø, serão utilizados caracteres grandes para texto. Se X estiver entre 41 e 8Ø os caracteres serão pequenos.

Os parâmetros de fábrica, quando o KIT 2+ é adquirido, são:

SCREEN Ø - 37 colunas
SCREEN 1 - 29 colunas

Ex:

```
1Ø SCREENØ
2Ø FOR N=1 TO 8Ø
3Ø WIDTH N
5Ø PRINT "Esta tela tem ";N;" colunas."
6Ø FOR T=Ø TO 3ØØ:NEXT
7Ø NEXT
```

CIRCLE, DRAW, LINE, PAINT, POINT, PRESET e PSET

Todas estas instruções passam a trabalhar com os novos valores das cores e coordenadas das telas gráficas.

Para obter os valores máximos das coordenadas, ver

instrução SCREEN.

Para saber as cores que podem ser utilizadas veja as instruções COLOR e SCREEN.

Nas SCREENs de 5 em diante, os desenhos podem ter uma cor para o traçado e serem preenchidos com outra diferente. A instrução PAINT passa a aceitar a seguinte sintaxe:

PAINT C1,C2

onde C1 é a cor que irá pintar a figura e C2, a cor do traçado ou borda da figura.

GETDATE X$ — Instrução

Obtém a data armazenada no relógio/calendário interno.

A data no relógio interno é transferida para a variável X$. O formato dos dados na variável é o seguinte:

DD/MM/AA - onde DD é o dia, MM o mês e AA o ano.

A data armazenada no relógio/calendário interno é atualizada mesmo com o computador desligado.

Ex:

```
GETDATE A$ : PRINT A$
```

SETDATE X$ — Instrução

Acerta a data no relógio interno.

X$ é uma expressão alfanumérica com o seguinte formato:

DD/MM/AA - onde: DD é o dia (1 a 31), MM é o mês (1 a 12) e AA é o ano.

A data é armazenada no relógio interno e será incrementada mesmo com o computador desligado.

Ex:

```
SETDATE "14/04/61"
GETDATE A$
PRINT A$
```

GETTIME X$ [,A] — Instrução

Obtém a hora armazenada no relógio interno.

A hora no relógio interno é transferida para a variável X$. O formato dos dados na variável é o seguinte: HH/MM/SS - onde HH é a hora, MM o minuto e SS o segundo.

A hora do relógio é atualizada mesmo com o computador desligado. Ex:

```
10 SCREEN 1
20 GETTIME A$
30 LOCATE 10,10
40 PRINT A$
50 GOTO 20
```

SETTIME X$ — Instrução

Acerta a hora do relógio interno.

X$ é uma sequência de caracteres com o seguinte formato: HH:MM:SS - onde HH é a hora (entre 00 e 23), MM o minuto (entre 00 e 59) e SS o segundo (entre 00 e 59).

Estes dados são guardados na memória do relógio interno que é mantido por baterias. Assim, mesmo com o computador desligado, eles serão preservados e atualizados. Ex:

```
10 SETTIME "13:30:20"
20 GETTIME A$
30 LOCATE 10,10
40 PRINT A$
50 GOTO 20
```

SETADJUST (X,Y) — Instrução

Ajusta a posição da tela para se adequar ao televisor ou monitor.

X deve ser um número inteiro entre -7 e 8. Quando

for negativo, a tela se deslocará para a esquerda. Quando for positivo, a tela se deslocará para a direita.

Y deve ser um número inteiro entre -7 e 8. Quando for negativo, a tela se deslocará para cima e quando for positivo, a tela se deslocará para baixo.

Os valores de X e Y são armazenados na memória do relógio. Assim, quando o computador for ligado, os parâmetros armazenados serão usados automaticamente. Ex:

```
10 SCREEN1:COLOR 15,4,7
20 PRINT "USE AS TECLAS DO CURSOR"
30 C=STICK(Ø)
40 X=X+(C=6 OR C=7 OR C=8)-(C=2 OR C=3 OR C=4)
50 Y=Y+(C=8 OR C=1 OR C=2)-(C=4 OR C=5 OR C=6)
60 IF X=-8 THEN X=-7
70 IF X=9 THEN X=8
80 IF Y=-8 Y=-7
90 IF Y=9 THEN Y=8
100 LOCATE 4,4
110 PRINT "X = ";X ,"Y = ";Y
120 SETADJUST (X,Y)
130 GOTO 30
```

SETBEEP X,Y — Instrução

Seleciona o tipo e o volume do BEEP.

X é o tipo de BEEP e deve ser um número inteiro entre 1 e 4.

Y é o volume do BEEP e deve estar entre 1 e 4. O número 1 é o volume mínimo e o 4 o volume máximo.

Os valores de X e Y são armazenados na memória do relógio interno. Assim, quando o computador for ligado, os parâmetros armazenados serão usados automaticamente.

O BEEP também pode ser gerado pressionando as teclas CONTROL+G.

```
10 FOR A=1 TO 4
20 FOR B=1 TO 4
30 SETBEEP A,B
40 FOR N=1 TO 10 : BEEP : NEXT
50 FOR T=Ø TO 500:NEXT
60 NEXT:NEXT
```

SETPASSWORD X$ — Instrução

Estabelece uma senha ou palavra de acesso ao computador.

X$ é uma sequência alfanumérica de até 255 caracteres.

A senha é armazenada na memória do relógio interno que é mantido por baterias. Assim, mesmo com o computador desligado, a senha continua armazenada.

Ao se ligar o computador, será apresentada a palavra "PASSWORD:" esperando que seja digitada a senha.

Caso a senha digitada não coincida com aquela armazenada, o computador continuará esperando a senha correta.

Para eliminar a senha, use a instrução SETTITLE"".

Só se pode utilizar uma das instruções "SETPASSWORD", "SETPROMPT" ou "SETTITLE" de cada vez. A última instrução executada será a armazenada pois todas utilizam a mesma área na memória do relógio interno.

Ex:

SETPASSWORD"TESTE"
desligue o computador e ligue-o novamente.
digite TESTE seguido da tecla RETURN

SETPROMPT X$ — Instrução

Estabelece um PROMPT ou palavra de aviso de espera de um comando.

X$ é uma sequência alfanumérica de até 6 caracteres.

O PROMPT é armazenado na memória do relógio interno que é mantido por baterias. Assim, mesmo com o computador desligado, ele será preservado.

O PROMPT normal do computador é "Ok". Quando este PROMPT é selecionado, a área na memória de relógio fica livre para ser usada pelas instruções: SETPASSWORD ou SETTITLE.

Só se pode utilizar uma das instruções

"SETPASSWORD", "SETPROMPT" ou "SETTITLE" de cada vez. A última instrução executada será a armazenada pois elas utilizam a mesma área na memória do relógio interno.
Ex: SETPROMPT "TESTE"

SETSCREEN — Instrução

Transfere os parâmetros da tela e do modo de trabalho do computador para a memória do relógio.

Os parâmetros armazenados são:

- Tipo de SCREEN de texto: Ø ou 1.
- Número de colunas (WIDTH): 1 a 8Ø.
- Número da cor do caractere: Ø a 15.
- Número da cor do fundo do caractere: Ø a 15.
- Número da cor da borda da tela: Ø a 15:
- Exibição das teclas de funções: ligada ou desligada.
- Click do teclado: ligado ou desligado.
- Tipo de impressora: MSX ou não MSX
- Velocidade dos dados para o gravador cassete 12ØØ ou 24ØØ bauds.
- Modo de exibição da tela: 1 a 4

Estes dados são armazenados na memória do relógio interno, mantido por baterias. Assim, mesmo com o computador desligado, eles serão preservados. Ex:

Para inicializar o computador em 8Ø colunas automaticamente:

SCREENØ : WIDTH8Ø : SETSCREEN

Desligue o computador e ligue-o novamente.

SETTITLE X$ [,Y] — Instrução

Armazena um nome que será mostrado na apresentação quando o computador for ligado.

X$ é uma sequência alfanumérica com até 6 caracteres.

Se X$ tiver exatamente 6 caracteres, o nome aparecerá na apresentação e o computador ficará

esperando que qualquer tecla seja pressionada, continuando a inicialização somente depois. Se tiver até 5 caracteres, o nome permanecerá na tela durante alguns segundos e, então, continuará a inicialização.

Y pode variar de 1 a 4 e seleciona qual combinação de cores será usada na apresentação quando o computador for ligado.

Estes dados são armazenados na memória do relógio interno que é mantido por baterias. Assim, mesmo com o computador desligado, os dados serão preservados.

Só se pode utilizar uma das instruções "SETPASSWORD", "SETPROMPT" ou "SETTITLE" de cada vez. A última instrução executada será a armazenada pois todas utilizam a mesma área na memória do relógio interno.

SETTITLE "" é usado para apagar os dados do SET PASSWORD.

Ex: SETTITLE "TESTE ",4

Deslígue o computador e ligue-o novamente.

CALL MEMINI [(X)] — Instrução

Prepara uma área da memória RAM para trabalhar como RAMDISK (disco em memória).

X é o tamanho do disco em memória em bytes e tem de ser um número inteiro entre 1023 e 32767, ou Ø (zero).

CALL MEMINI (Ø) cancela o modo RAMDISK.

Quando se executa a instrução CALL MEMINI, todo conteúdo do disco em memória será limpo.

No lugar da palavra CALL pode-se utilizar o sinal de sublinhado (_). Por exemplo: _MEMINI

A instrução CALL MEMINI deve ser executada antes das instruções: CALL MFILES, CALL MNAME ou CALL MKILL.

As seguintes instruções podem ser utilizadas após a instrução CALL MEMINI para se trabalhar com arquivos em RAMDISK: SAVE, LOAD, RUN, MERGE, OPEN, CLOSE, PRINT#, PRINT#USING, INPUT#, LINEINPUT#,EOF, LOC, LOF.

A RAMDISK deve ser tratada como se fosse um disquete normal. O único detalhe é que o nome do dispositivo deve ser MEM:.

Ex: _MEMINI

```
10 PRINT "TESTE "; : GOTO 10
SAVE "MEM:TESTE.BAS"
NEW
LIST
LOAD "MEM:TESTE.BAS"
LIST
RUN
```

CALL MFILES — Instrução

Lista na tela os nomes dos arquivos armazenados na RAMDISK e quantos bytes ainda estão livres para uso.

No lugar da palavra CALL pode-se utilizar o sinal de sublinhado (_). Por exemplo: _MFILES

Este comando só será aceito depois da execução da instrução CALL MEMINI. Caso a instrução CALL MEMINI não tenha sido executada ou a RAMDISK tenha sido eliminada pela instrução CALL MEMINI (0), será emitida a mensagem de erro "disk offline".

Ex: _MEMINI

```
10 PRINT "TESTE "; : GOTO 10
SAVE "MEM:TESTE.BAS"
NEW
_MFILES
LIST
LOAD "MEM:TESTE.BAS"
LIST
RUN
```

CALL MKILL ("nome do arquivo") — Instrução

Apaga um arquivo armazenado na RAMDISK.

No lugar da palavra CALL pode-se utilizar o sinal de sublinhado (_). Por exemplo: _MFILES

Este comando só será aceito depois da execução da instrução CALL MEMINI. Caso a instrução CALL MEMINI não tenha sido executada ou a RAMDISK tenha sido eliminada pela instrução CALL MEMINI (0), será emitida a mensagem

de erro "disk offline".
Ex: _MEMINI

```
10 PRINT "TESTE "; : GOTO 10
SAVE "MEM:TESTE.BAS"
_MFILES
_MKILL "TESTE.BAS"
_MFILES
```

CALL MNAME("nome arquivo 1" as "nome arq 2") Instrução

Troca o nome de um arquivo gravado na RAMDISK.

O "nome do arquivo 1" é o nome já gravado na RAMDISK e o "nome do arquivo 2" é aquele que substituirá o antigo.

No lugar da palavra CALL pode-se utilizar o sinal de sublinhado (_). Por exemplo: _MNAME

Este comando só será aceito depois da execução da instrução CALL MEMINI. Caso a instrução CALL MEMINI não tenha sido executada ou a RAMDISK tenha sido eliminada pela instrução CALL MEMINI (0), será emitida a mensagem de erro "disk offline".

Ex: _MEMINI

```
10 PRINT "TESTE "; : GOTO 10
SAVE "MEM:TESTE.BAS"
_MFILES
_MNAME ("TESTE.BAS" AS "PROG.BAS")
_MFILES
```

PAD X Instrução

Retorna o estado da TOUCHPAD, LIGHTPEN, MOUSE ou TRACKBALL que estejam ligados às entradas de JOYSTICK.

X deve ser um número inteiro entre 0 e 19 e tem o seguinte significado:

Se X estiver entre 0 e 3 presume-se que há uma TOUCHPAD ligada na tomada do JOYSTICK 1.

Se X estiver entre 4 e 7 presume-se que há uma TOUCHPAD ligada na tomada do JOYSTICK 2.

Se X estiver entre 8 e 11 presume-se que tenha uma LIGHTPEN ligada na tomada do JOYSTICK.

Se X estiver entre 12 e 15 presume-se que tenha um MOUSE ou TRACKBALL ligado na tomada do JOYSTICK 1.

Se X estiver entre 16 e 19 presume-se que tenha um MOUSE ou TRACKBALL ligado na tomada do JOYSTICK 2.

Nos casos acima se X for igual a:

Ø ou 4 - retorna o estado da mesa da TOUCHPAD: Ø se não foi tocada ou -1 se foi tocada.

1 ou 5 - retorna a coordenada X do ponto onde a mesa foi tocada.

2 ou 6 - retorna a coordenada Y do ponto onde a mesa foi tocada.

3 ou 7 - retorna o estado do botão da TOUCHPAD. O resultado será: Ø (botão não pressionado) ou -1 (botão pressionado).

8 - retorna o estado da LIGHTPEN: Ø se as coordenadas não estiverem prontas ou -1 se as coordenadas estiverem prontas.

9 - retorna a coordenada X da posição da LIGHTPEN.

1Ø - retorna a coordenada Y da posição da LIGHTPEN

11 - retorna o estado do botão da LIGHTPEN: Ø (botão não pressionado) ou -1 (botão pressionado)

12 ou 16 - retorna o estado do MOUSE ou TRACKBALL: é sempre -1

13 ou 17 - retorna a coordenada X do MOUSE ou TRACKBALL.

14 ou 18 - retorna a coordenada Y do MOUSE ou TRACKBALL.

15 ou 19 - retorna sempre Ø.

A leitura do estado dos botões do MOUSE ou TRACKBALL é feita com a instrução STRIG.

Antes de ler as coordenadas do MOUSE ou TRACKBALL é obrigatória a execução da função PAD (12) ou PAD (16) dependendo da tomada onde o MOUSE ou TRACKBALL estejam ligados.

Antes de ler as coordenadas da LIGHTPEN, verifique se a função PAD(8) retorna o valor -1.

A função PAD só terá validade se algum dos acessórios (TOUCHPAD, LIGHTPEN, MOUSE ou TRACKBALL) estiver ligado ao computador.

- TOUCHPAD é uma pequena prancheta usada para desenhos com sensores em toda sua extensão que, ao serem tocados, transmitem ao computador as coordenadas X,Y. Para acompanhar os traços do desenho é usada uma caneta com dois botões: um como nos JOYSTICKS e outro que é acionado quando a caneta toca a superfície da prancheta.

- LIGHTPEN é uma caneta com um sensor de luz na ponta que, quando encostada na tela do monitor ou televisor, transmite ao computador a sua posição em coordenadas X,Y. A Lightpen possui dois botões: um como nos JOYSTICKS e outro que é o sensor quando cruza com um raio luminoso na tela.

- MOUSE é um controlador manual com uma esfera na sua parte inferior e dois botões como nos JOYSTICKS. A movimentação do MOUSE sobre qualquer superfície faz com que a esfera movimente duas rodas perfuradas, uma para o eixo X e outra para o eixo Y. Estas rodas geram uma sequência de pulsos que são transformados em coordenadas pelo computador.

- TRACKBALL é semelhante ao MOUSE, porém, a esfera fica na parte superior do controlador. Ao contrário do Mouse, sua esfera é controlada pela mão do operador que a faz girar para o lado desejado.

PDL (X) — Função

Retorna o estado de PADDLES ligados ao computador.

Até 12 (doze) PADDLES podem ser ligados às entradas de JOYSTICK (6 na entrada A e 6 na B).

Se X for 1, 3, 5, 7, 9 ou 11 pressume-se que os PADDLES estejam ligados na entrada do JOYSTICK-1 (A). Se for 2, 4, 6, 8, 10 ou 12 pressume-se que estejam ligados na entrada do JOYSTICK-2 (B).

Esta função retorna um valor entre Ø e 255.

PADDLEs são controles rotativos utilizados em jogos ou programas que têm movimentos em duas direções (direita e esquerda por exemplo).

A função PDL(X) só é válida com um ou mais PADDLES ligados ao micro.

Ex: 1Ø PRINT PDL(1) : GOTO1Ø

BASE (X) ou BASE (X) = Y — Função ou Instrução

Mostra o endereço do primeiro byte das tabelas do processador de vídeo (VDP) ou escreve um endereço inicial nos registradores das tabelas do VDP.

(X) deve ser sempre uma expressão de números inteiros entre Ø e 44 que, dependendo da SCREEN utilizada, tem o seguinte significado:

SCREEN	Ø	1	2	3	4	5	6	7	8
TABELA NOMES	Ø	5	1Ø	15	2Ø	25	3Ø	35	4Ø
TABELA CORES	-	6	11	16	21	26	31	36	41
TABELA CARACTERES	2	7	12	17	22	27	32	37	42
TABELA ATRIB SPRITES	-	8	13	18	23	28	33	38	43
TABELA FORMATO SPRITES	-	9	14	19	24	29	34	39	44

Quando se lê um endereço com a função BASE (X), o valor de X pode variar entre Ø e 44.

Para se escrever um valor nos registradores do VDP com a função BASE (X) = Y, o valor de X só pode variar entre Ø e 19, ou seja, somente as SCREENs Ø, 1, 2 e 3 são afetadas. Quando se troca um endereço na SCREEN 3, os endereços das SCREENs 4, 5, 6, 7 e 8 também são alterados.

Os dados da SCREEN 8 valem também para as SCREENs 1Ø, 11 e 12. Ex:

```
100 'Enche duas telas
110 '
120 SCREEN0:WIDTH40:KEYOFF
130 '
140 FORA=0TO959:PRINT"-";:NEXT
150 BASE(0)=1024:SCREEN0
160 FORA=0TO959:PRINT"+";:NEXT
```

```
170 '
180 'Alterna as páginas de texto
190 '
200 BASE(0)=0
210 FORT=0TO100:NEXT
220 BASE(0)=1024
230 FORT=0TO100:NEXT
240 GOTO200
```

Outro exemplo: PRINT BASE(0)

OBS: Use esta função ou instrução somente quando estiver bem familiarizado com o modo de trabalho do processador de vídeo (VDP).

VDP (X) ou VDP (X) = Y — Função ou Instrução

Verifica ou altera o conteúdo dos registradores do processador de vídeo (VDP).

X é um número inteiro e pode variar entre:

-9 e -1 - registradores de STATUS 1 a 9.
0 e 8 - mesmos do MSX1.
9 e 24 - registradores de 8 a 23.
33 e 47 - registradores de 32 a 46.

Esta instrução só deve ser utilizada quando se estiver bem familiarizado com o funcionamento do processador de vídeo (VDP).

Ex:

```
10 FOR R=-9 TO 8
20 A=VDP(R)
30 PRINT "Registro: "R;" = ";RIGHT$("0000000"+BIN$(A),8)
40 NEXT
```

F M - MUSIC

APRESENTAÇÃO

O FM-Music, residente no cartucho MSX 2+ FM (e também vendido como cartucho separado) é capaz de sintetizar o som de diversos instrumentos musicais, utilizando-se de 2 osciladores multi-timbrais e mais os sons de 5 instrumentos de percussão. Nele estão disponíveis 9 canais que podem tocar até 9 instrumentos ao mesmo tempo sem utilizar a bateria.

Contudo, se os 5 canais da bateria eletrônica forem acionados, ficam ainda 6 canais simultâneos para outros instrumentos. Os 3 canais do PSG (chip de som original do MSX) continuam ativos, podendo ser tocados junto com os canais do FM, resultando em até 14 canais simultâneos.

O Music-Basic tem armazenado os sons de 64 instrumentos musicais sendo 63 pré-definidos e 1 programável. Os pré-definidos (de Ø a 62) são os mais utilizados, como piano, flauta, clarineta, violino, cordas, trompete, xilofone, etc.

O som desses instrumentos pode ser alterado através de rotinas em Assembler ou Poke's em Basic. A programação do instrumento 63 é feita através de parâmetros transmitidos aos registradores dos osciladores. É necessário, portanto, um certo conhecimento de sonorização para sintetizar com perfeição outros tipos de instrumentos.

O FM pode ser programado tanto em Basic quanto em Assembler. Em Basic através de comandos expandidos, tais como Call Music, Call Voice, entre outros, que serão descritos neste manual. Em Assembler através do rotinas específicas em linguagem de máquina (um processo que, pelo nível de complexidade, não será explicado neste manual).

Descrição da sintaxe usada no Capítulo FM-Music.

NOME [<P>]

Descrição da utilização do comando.
NOME (Parâmetro1, Parâmetro2, [Parâmetro3]...)
Os parâmetros virão entre parênteses ( ).
Os itens entre colchetes [ ] indicam que esse parâmetro depende de outros comandos.
OBSERVAÇÃO: A palavra CALL pode ser substituída por um "_" (underline). Ex: CALL MUSIC = _MUSIC

CALL MUSIC (<BT>,<Ø>,<C1>,<C2>,<C3>,<C4>,...,[C8],[C9])

Define os canais de som que serão utilizados, bem como se será ou não usada a bateria. Com este comando é possível também a concatenação de canais, respeitando-se o limite disponível (6 canais com bateria ou 9 canais sem bateria).

O primeiro parâmetro <BT> pode ser Ø ou 1. Se Ø, não será utilizada a bateria. Se 1, os 3 canais de bateria serão ativados.

O segundo parâmetro deve ser obrigatoriamente Ø. Qualquer valor diferente de Ø gerará uma mensagem de erro.

Os parâmetros de <C1> a <C9> referem-se ao chaveamento de canais de som. Esses parâmetros devem conter um número de 1 a 6 (ou de 1 a 9 dependendo da bateria). Esse número indicará o máximo de vozes que será tocado ou mixado nesse canal.

Ex:

```
CALL MUSIC (1,Ø,1,1,1,1,1,1).
```

Com esse comando será acionada a bateria e cada um dos 6 canais de som disponíveis poderá tocar uma voz (instrumento) diferente.

```
CALL MUSIC (Ø,Ø,2,1,1,1)
```

Neste exemplo, não será utilizada a bateria e teremos 3 dos 9 canais disponíveis, sendo que o primeiro

(que contém o nº 2) poderá tocar até dois instrumentos mixados.

É importante observar que a concatenação de canais (como no exemplo acima) só pode ser feita respeitando-se o número máximo de canais do FM. Sendo assim, a soma dos parâmetros (apenas dos canais) não deverá ultrapassar o limite de 6 (com bateria) ou 9 (sem bateria). Nesse caso:

```
CALL MUSIC (1,Ø,2,2,2,2)
```

Gerará uma mensagem de erro.

CALL VOICE (<@I1>,<@I2>,<@I3>,<@I4>,......,[@I8],[@I9])

Define os instrumentos que serão utilizados em cada canal.

Estão disponíveis 64 instrumentos. Os de Ø a 62 são os instrumentos normais (pré-definidos) e o 63 é o instrumento programável que não possui som algum.

Os labels de <I1> a <I9> deverão ser substituídos pelos números dos instrumentos. Deverão ser sempre precedidos por "@".

Ex:

```
CALL VOICE (@ØØ,@Ø3,@43)
```

Esse comando definiu a utilização de PIANO no canal 1, FLAUTA no canal 2 e VIOLINO no canal 3.

```
CALL VOICE (@Ø2,@16,@Ø1,@Ø1,@Ø1)
```

Nesse exemplo ficou definida a utilização de CORDAS no canal 1, VIBRAPHONE no canal 2, e PIANO II nos canais 3, 4 e 5.

LEMBRE-SE que o número de canais depende da instrução CALL MUSIC e que, caso seja definido algum canal com concatenação, o comando CALL VOICE deverá conter os instrumentos que serão usados em cada canal concatenado. Assim, ao se executar:

```
CALL MUSIC (Ø,Ø,3):CALL VOICE (@ØØ,@36,@Ø2):PLAY#2,
"C"
```

Você irá tocar a nota Dó (C) com PIANO, FLAUTA e CORDAS mixados, no primeiro (e único) canal.

OBS: No APÊNDICE C, há uma relação com todos os instrumentos pré-definidos.

PLAY# <n>,<C1>,<C2>,.....<C8>,<C9>,<PSG1>,<PSG2>,<PSG3>

Toca a música de acordo com os parâmetros nele definidos (nota, volume, duração tempo, etc...).

O comando PLAY original do MSX sofreu uma pequena alteração: recebeu um parâmetro a mais, o #<n>, para indicar que estamos nos reportando ao FM.

PLAY#<n>,<C1>,<C2>,<C3>,<C4>,<C5>,<C6>,<C7>,<C8>,<C9>,<PSG1>,<PSG2>,<PSG3>

Esta primeira sintaxe se refere à utilização sem bateria. Nesse caso, teremos 9 canais disponíveis, (<C1> a <C9>), mais os 3 canais do PSG, (<PSG1> a <PSG3>)

ou

PLAY#<n>,<C1>,<C2>,<C3>,<C4>,<C5>,<C6>,<BT>,<PSG1>,<PSG2>,<PSG3>

Nesta segunda sintaxe utilizamos a bateria (lembre-se que isso depende do comando CALL MUSIC). Nesse caso, teremos 6 canais (<C1> a <C6>), a bateria (<BT> e os 3 canais do PSG (<PSG1> a <PSG3>). Note que a bateria não faz parte dos canais de música. Ela será definida sempre depois do último canal musical. Assim, se fizermos:

CALL MUSIC (1,Ø,1,1,1,1)

Teremos que definir a bateria no quinto canal do comando PLAY:

PLAY#2,<C1>,<C2>,<C3>,<C4>,<BT>

O <n> pode ser substituído por Ø, 2 ou 3. Se for 2 ou 3, valem as sintaxes acima explicadas. Se for Ø, indica que estaremos trabalhando apenas com o PSG. Assim, o comando PLAY não poderá ultrapassar 3 canais:

PLAY#Ø,<PSG1>,<PSG2>,<PSG3>

Todos os labels acima mostrados (<C1> a <C9>, <BT>, <PSG1> a <PSG3> deverão ser substituídos por parâmetros colocados entre " " (aspas).

Parâmetros para SOM:

@<n> - Muda de instrumento dentro do PLAY. <n> pode variar de Ø a 63 (igual ao CALL VOICE). O default é Ø.

V<n> - Altera o volume do som. <n> pode variar de Ø a 15 e o default é 8.

T<n> - Muda o tempo da música (andamento). <n> pode ser de 32 a 255, default e 12Ø.

L<n> - Altera a duração de cada nota. <n> varia de 1 a 64. O default é 4.

O<n> - Define a oitava nota utilizada. <n> pode variar de 1 a 8 sendo que o default é 4.

Q<n> - Muda o "decay" (finalização) das notas. <n> pode variar de 1 a 64. default é 8.

N<n> - Toca uma nota especificada pelo seu número. <n> pode ser de Ø a 96.

R<n> - Pausa por "n" tempos. <n> varia de 1 a 64. O default é 4.

"."- Quando colocado ao lado da nota, determina que esta será tocada por mais 1/2 tempo de seu valor original.

@V<n>- Assim como o V, esse comando altera o volume do som mas cobre uma faixa de valores maior do que o V.

<n> pode ser de Ø a 127, o default é 127.

@W<n> - Prolonga a execução da última nota por <n> tempos. <n> pode ser de 1 a 64.

"&"- Faz a "junção de 2 notas de mesmo valor.

">"- Aumenta 1 (uma) oitava.

"<"- Diminui 1 (uma) oitava.

S<n> e M<n> - Comandos para geração de envelopes. Só funcionam com o PSG. Para maiores detalhes, consulte o manual de BASIC MSX.

NOTAS MUSICAIS: C (dó), D (ré), E (mi), F (fá), G (sol), A (lá), B (si). As notas podem estar acompanhadas de um número (de 1 a 64) indicando sua duração. Se não especificado, é assumido o valor do parâmetro L.

"#" ou "+" - Acompanhado da nota, indica que esta será SUSTENIDA (1/2 tom acima).

"-" - Indica que a nota será BEMOL (1/2 tom abaixo).

Y<n1>,<n2> - Altera a definição de alguns instrumentos pré-definidos.

Parâmetros para BATERIA:

@A<n>- Indica o volume dos instrumentos tocados em primeiro plano (com ênfase).

<n> pode ser de Ø a 15, o default é 12.

V<n> - Indica o volume dos instrumentos tocados em segundo plano. Idêntico ao V do SOM.

T<n> - Idêntico ao T do SOM.

@V<n>- Define o volume geral (primeiro e segundo

planos). Idêntico ao @V do SOM.

"!"- ênfase no instrumento (este é tocado no primeiro plano)

R<n> - Idêntico ao R do SOM.

Y<n1>,<n2> - Permite alterar o som dos instrumentos da bateria. (maiores detalhes, consulte o comando CALL AUDREG).

INSTRUMENTOS DE BATERIA:

B - Bumbo

H - Hi-Hat

C - Chimbau

S - Caixa

M - Tom Tom

Os instrumentos de bateria devem, obrigatoriamente, vir acompanhados do valor de duração (de 1 a 64). É possível tocar até 5 instrumentos ao mesmo tempo, colocando um ao lado do outro e, no final, o valor de duração.

Ex:

```
CALL MUSIC (1)
PLAY#2,"BC4"
```

Tocará o BUMBO e o CHIMBAU juntos. Quando colocados dessa forma, os sons dos instrumentos aparecem em segundo plano. Para tocar um instrumento em primeiro plano, deve-se colocar o sinal "!" logo após. Ex:

```
CALL MUSIC (1)
PLAY#2,"B!H4"
```

Tocará o BUMBO em primeiro plano junto com o HI-HAT em segundo plano. É importante ressaltar que os instrumentos tocados no primeiro plano somente terão ênfase se o volume sonoro do segundo plano for menor. Ex:

```
CALL MUSIC (Ø,Ø,1,1,1)
PLAY
#2,"V15O4@ØØC1G1F1","V15O4@ØØE1B1A1","V15O4@ØØG1>D1C1"
```

Nesse exemplo, serão tocados 3 acordes: Dó maior, SOL maior e FÁ maior com som de PIANO (@ØØ).

```
CALL MUSIC (1,Ø)
PLAY #2,"@A15VØB!H8H8S!H8B!H8B!H8H8S!H8C!H8"
```

Aqui é possível ouvir um ritmo de bateria (ROCK). Note como foi utilizada a junção de instrumentos de bateria.

```
CALL MUSIC (Ø,Ø,3)
POKE &HFA3C,4Ø:POKE &HFA4C,8Ø
CALL VOICE (@53,@53,@53)
PLAY #2,"V15O4C1<G1A1F1"
```

Neste último exemplo, serão tocados algumas notas com o instrumento 53 (HARDROCK) apresentando uma leve reverberação que será explicada no APÊNDICE B.

CALL PITCH (<n>)

Altera a velocidade tonal da música. É útil quando se deseja acertar a afinação dos instrumentos pois atua sobre todos os canais.

<n> pode ser um número de 41Ø a 459 (41Ø mais lento, 459 mais rápido). O default é 44Ø.

Ex:

```
CALL MUSIC (Ø,Ø,1)
PLAY #2,"V15C"
CALL PITCH (459)
PLAY #2,"V15C"
CALL PITCH (41Ø)
PLAY #2,"V15C"
```

CALL TRANSPOSE (<n>)

Semelhante ao PITCH, porém, cobre uma faixa maior de valores.

<n> pode ser qualquer número entre -12799 e 12799 sendo a cada 1ØØ valores aumentado (ou diminuido) meio tom. O default é Ø.

Ex:

```
CALL MUSIC (Ø,Ø,1)
CALL TRANSPOSE (Ø)
PLAY #2,"V15C"
CALL TRANSPOSE (-5ØØ)
PLAY #2,"V15C"
CALL TRANSPOSE (5ØØ)
PLAY #2,"V15C"
```

Digitando esses comandos você notará a mudança entre diferentes valores do TRANSPOSE.

CALL TEMPER (<n>)

Serve para afinar os instrumentos.
<n> deve ser um número entre Ø e 21.

Ex:

```
CALL MUSIC (Ø,Ø,1)
CALL TEMPER (Ø)
PLAY #2,"@12L8CDEFGAB"
CALL TEMPER (21)
PLAY #2,"@12L8CDEFGAB"
```

Note que o som gerado pelo segundo exemplo está "desafinado".

CALL PLAY (<nc>,<var>)

O comando CALL PLAY indica se os canais de FM estão ou não ativos (tocando).

<nc> é o número do canal que se deseja saber se está ou não tocando. Se <nc> for Ø será possível saber se existe algum canal ativo. Enquanto um deles estiver tocando o valor devolvido será -1, caso contrário, será Ø. Com <nc> de 1 a 9, você saberá apenas o estado (ativo ou não) do canal escolhido.

<var> é a variável numérica onde será colocado o valor devolvido. Ela conterá Ø caso o(s) canal(is) não esteja(m) tocando e, -1 se estiver(em) tocando.

Ex:

```
CALL PLAY (3,A):PRINT A
```

Lê o estado do canal 3 e retorna o resultado para a variável "A". Como não tocamos nota nenhuma, o valor devolvido será Ø.

```
CALL MUSIC (Ø,Ø,1)
PLAY #2,"V15C":CALL PLAY (Ø,A):PRINT A
```

Esse exemplo irá tocar uma nota enquanto lê o estado do canal 1. Obviamente o valor devolvido será -1.

CALL STOPM

Determina uma parada na música no ponto em que se encontra. Como sabemos, o processador Z8Ø trabalha a uma velocidade maior que o processador de áudio (PSG) e o FM. Portanto, quando executamos uma ou várias notas, algumas vezes, o controle é devolvido para o BASIC antes mesmo de haver terminado a sequência de notas, o que faz com que os próximos comandos (no caso de um programa), após a instrução PLAY, sejam executados antes do término da música.

Ex:

```
10 CLS:LOCATE1Ø,1Ø:PRINT"EXEMPLO SEM STOPM"
2Ø CALL MUSIC (Ø,Ø,1)
3Ø PLAY#2,"V15@41O4CDEFGAB>CDEFGAB"
4Ø CLS
```

Ao executar esse programa você notará que a tela será limpa antes de terminar a música.

```
1Ø CLS:LOCATE1Ø,1Ø:PRINT"EXEMPLO COM STOPM"
2Ø CALL MUSIC (Ø,Ø,1)
3Ø PLAY#2,"V15@41O4CDEFGAB>CDEFGAB"
4Ø CALL STOPM
5Ø CLS
```

Nesse exemplo a tela será limpa simultaneamente à interrupção da música.

CALL BGM (<n>)

Sincroniza o processador Z8Ø com os processadores de áudio FM e PSG, evitando que os comandos, após uma instrução PLAY, sejam executados antes do término da música.

<n> pode ser Ø ou 1. Se for 1, o recurso estará desativado. Dessa forma o processador Z8Ø terminará sempre primeiro. Se for Ø, o recurso será ativado e os processadores executarão suas tarefas juntos. O default é 1.

Ex:

```
10 CLS:LOCATE10,10:PRINT"EXEMPLO SEM BGM"
20 CALL MUSIC (0,0,1):CALL BGM (1)
30 PLAY#2,"V15@4104CDEFGAB>CDEFGAB":CLS
```

Neste exemplo, a tela será limpa antes da música terminar.

```
10 CLS:LOCATE10,10:PRINT"EXEMPLO COM BGM"
20 CALL MUSIC (0,0,1):CALL BGM (0)
30 PLAY#2,"V15@4104CDEFGAB>CDEFGAB":CLS
```

Neste outro exemplo, a tela só será limpa após todas as notas terem sido executadas.

CALL AUDREG (<n1>,<n2>)

Altera o som dos instrumentos da bateria.

<n1> deverá ser um número com os valores abaixo:

de 00 a 07;
de 14 a 24;
de 32 a 40;
de 48 a 56;

Esse número indicará o registrador a ser alterado. Valores de 8 a 13, de 25 a 31, de 41 a 47, e acima de 39, geram mensagem de erro. Note que essa numeração é diferente da numeração normal dos instrumentos.

<n2> pode ser qualquer número inteiro, entre 0 e 255, e indicará o valor do atributo para o registrador.

Ex:

```
CALL MUSIC (1)
PLAY #2,"@A15V0B!H8H16B!16S!H8B!H8H8B!H8S!H8C!8"
CALL AUDREG (24,00)
PLAY #2,"@A15V0B!H8H16B!16S!H8B!H8H8B!H8S!H8C!8"
```

Com este exemplo, você notará a mudança no som do HI-HAT e do CHIMBAU.

```
10 CALL MUSIC (1)
20 FOR F=0 TO 255
30 CALL AUDREG (23,F)
40 PLAY #2,"V15S16"
50 NEXT F
```

Neste exemplo, é possível perceber a variação no som da CAIXA.

CALL VOICECOPY (@<ins>,<var>%)

Copia os valores dos registradores de som (definição) de cada instrumento para uma variável de precisão dupla. Permite também que passemos uma definição de instrumento (que são as informações necessárias para a composição do som) contida numa variável para os registradores do instrumento 63 (programável).

CALL VOICECOPY (@<ins>,<var>%)

A primeira sintaxe demonstra como colocar a definição de um instrumento (indicado por <ins>, de Ø a 63) numa variável de precisão dupla (indicada por <var>, seguida do símbolo de porcentagem).

Alguns instrumentos, por motivos desconhecidos, não são visualizados (geram mensagem de erro).

CALL VOICECOPY (<var>%,@63)

Neste outro exemplo, o instrumento 63 receberá os valores contidos na variável <var>. Os valores dos registradores estão divididos em 16 bytes de 16 bits cada, portanto, deverão ser tratados em blocos de 16 bytes (isso será melhor explicado no APÊNDICE A).

Ex:

```
10 CALL MUSIC:Y=Ø
20 CLEAR 1ØØØ:DIM A%(15)
3Ø CALL VOICECOPY (@1,A%)
4Ø FOR X=Ø TO 3
5Ø PRINT RIGHT$("ØØØØ"+HEX$(A%(X+Y)),4)" ";
6Ø NEXT X:Y=Y+4:PRINT:IF Y=16 THEN END ELSE 4Ø
```

No exemplo acima é possível ver na tela a definição hexadecimal do instrumento 1 (PIANO II).

```
1Ø CALL MUSIC (Ø,Ø,3)
2Ø CLEAR 2ØØØ:DIMA%(15)
3Ø FOR I=Ø TO 15
4Ø READ A$:A%(I)=VAL("&H"+A$)
5Ø NEXT I
```

```
60 DATA 6847,736F,2074,2020
70 DATA 0000,00FF,0000,0000
80 DATA 1572,01AA,0000,0000
90 DATA 0131,02A0,0000,0000
100 CALL VOICECOPY (A%,@63):POKE &HFA3C,40:POKE &HFA3C,80
110 PLAY#2,"V15@63O3G1D1F1C1"
```

Neste outro exemplo o instrumento 63 recebeu uma definição (contida nas linhas DATA), e foi acionado (tocado).

OBS: Somente o instrumento 63 pode receber uma definição do usuário. Os outros instrumentos, que são pré-programados, geram uma mensagem de erro.

APÊNDICE A - PROGRAMAÇÃO DE INSTRUMENTOS

O FM possui 64 instrumentos sendo um deles, o 63, programável. Mas existem algumas dificuldades para se gerar o som de um instrumento. Uma delas é saber o que faz cada um dos 16 registradores do FM. Abaixo estão especificados cada um dos bytes constantes da definição de um instrumento.

```
1) 0000,0000,0000,0000 -> Nome do instrumento (opcional)
2) 0000,0000,0000,0000 -> Afinação e envelope
3) 0000,0000,0000,0000 -> Dados 1º oscilador (CARRIER)
4) 0000,0000,0000,0000 -> Dados 2º oscilador (MODULATE)
```

Fig. 1

No tópico 1 temos um espaço de 8 bytes reservado para o nome do instrumento. O nome de todos os instrumentos pré-programados é definido nesse primeiro grupo de bytes. Não é obrigatório mas, se colocado, deverá ser feito no padrão ASCII em hexadecimal.

No tópico 2 é definida a afinação inicial (default) do instrumento. No primeiro byte se faz o ajuste grosso e no segundo byte, o ajuste fino. O terceiro byte é, pelo menos até a confecção deste manual, inútil. O quarto byte define o valor do gerador de envelope (formato da amplitude do som) para mixagem dos dois osciladores. Os quatro bytes restantes (5, 6, 7 e 8) têm suas funções desconhecidas.

No tópico 3 temos os bytes referentes ao 1º oscilador. O primeiro define o nível de graves no som. O segundo byte encontra-se subdividido em 4 partes: o primeiro número indica a vibração (para efeitos vibro e trêmolo) e o segundo, a tonalidade do som. O terceiro byte demonstra o prolongamento do som em relação ao gerador de envelopes e o quarto byte, o ataque do 1º oscilador. Também nesse caso os últimos 4 bytes são inúteis.

No tópico 4 encontra-se o 2º oscilador. Existem poucas diferenças entre os dois osciladores. O primeiro byte tem a mesma função que no tópico anterior com exceção de seu ajuste, ligeiramente mais fino. O segundo byte está subdividido em 3 partes: o primeiro número indica a vibração (que, dependendo de alguns fatores, pode não funcionar) e o segundo, a tonalidade deste oscilador. O terceiro byte é diferente: demonstra o volume dos dois osciladores. O último byte deste oscilador indica o ataque. Veja a Fig.2

```
1a. linha -> 00 00 , 00 00 , 00 00 , 00 00
                 Nome do instrumento
2a. linha -> 00 00 , 00 00 , 00 00 , 00 00
             |  |    |  |
             |  |    |  +-> Envelope
             |  |    +----> Inútil
             |  +---------> Ajuste fino da afinação
             +------------> Ajuste grosso da afinação
3a. linha -> 00 00 , 00 00 , 00 00 , 00 00
             |  ||   |  |
             |  ||   |  +-> Ataque
             |  ||   +----> Decaí
             |  |+--------> Tonalidade
             |  +---------> Efeito Vibro/Trêmolo
             +------------> Timbre
4a. linha -> 00 00 , 00 00 , 00 00 , 00 00
             |  ||   || |
             |  ||   || +-> Ataque
             |  ||   |+---> Prolongamento (éco)
             |  ||   +----> Volume
             |  |+--------> Tonalidade
             |  +---------> Efeito Vibro/Trêmolo
             +------------> Timbre               (Fig. 2)
```

OBS: Os últimos bytes de cada linha (exceto a primeira), que não têm função, podem assumir qualquer valor.

Os valores deverão ser sempre colocados em hexadecimal.

No APÊNDICE C você terá uma relação de todos os instrumentos.

APÊNDICE B - DICAS E MACETES

O FM Music se caracteriza principalmente pela facilidade de programação tanto em BASIC quanto em ASSEMBLER. Neste apêndice apresentaremos alguns truquezinhos que produzem bons efeitos musicais.

GERAÇÃO DE ECO OU REVERBERAÇÃO:

O som dos instrumentos no FM Music são originalmente compostos por apenas 2 osciladores. São considerados sons puros como se fossem extraídos direto do instrumento (que está sendo simulado), ou seja, não há a entonação timbral como numa música de estúdio. Para solucionar este problema é necessário adicionar mais um canal ao que desejamos tocar e colocar um POKE que vai alterar gradualmente o timbre da nota que estivermos tocando, dando a impressão de uma pequena reverberação.
Ex: Execute as linhas abaixo:

```
10 CALL MUSIC(Ø,Ø,1)
20 PLA┘#2,"@12V15L4CDEFGAB"
```

Selecionamos o instrumento 12 (Guitarra) e tocamos as notas de Dó a SI. Agora, execute essas linhas:

```
10 CALL MUSIC(Ø,Ø,2):POKE&HFA3C,4Ø
20 PLAY#2,"@12V15L4CDEFGAB"
```

Fizemos a mesma coisa só que, agora, o som ficou mais polido. Por que? Porque o POKE acima (POKE&HFA3C,4Ø) altera para 4Ø o timbre das notas tocadas no 2º canal (note que no segundo exemplo foram definidos 2 canais). Esse endereço de memória (&HFA3C) é uma variável utilizada pelo FM para armazenamento do conteúdo definido pela instrução CALL PITCH (que também faz essa alteração só que em todos os canais ao mesmo tempo).

A seguir estão relacionadas algumas variáveis de todos os canais:

&HFA2C - PITCH CANAL 1 ¦ &HFA2D - TRANSPOSE CANAL 1
&HFA3C - PITCH CANAL 2 ¦ &HFA3D - TRANSPOSE CANAL 2
&HFA4C - PITCH CANAL 3 ¦ &HFA4D - TRANSPOSE CANAL 3
&HFA5C - PITCH CANAL 4 ¦ &HFA5D - TRANSPOSE CANAL 4
&HFA6C - PITCH CANAL 5 ¦ &HFA6D - TRANSPOSE CANAL 5
&HFA7C - PITCH CANAL 6 ¦ &HFA7D - TRANSPOSE CANAL 6
&HFA8C - PITCH CANAL 7 ¦ &HFA8D - TRANSPOSE CANAL 7
&HFA9C - PITCH CANAL 8 ¦ &HFA9D - TRANSPOSE CANAL 8
&HFAAC - PITCH CANAL 9 ¦ &HFAAD - TRANSPOSE CANAL 9

Um outro recurso para que o som saia com eco consiste em utilizar 2 canais tocando uma nota em cada. Ex:Suponhamos tocar as seguintes notas:

"@12V15C4D8A4E16F16"

Podemos separar em dois canais:

"@12V15C4R8A4R16F16","@12V15R4D8R4E16R16"

Assim, teremos uma nota em cada canal gerando um bonito efeito de eco. Para que isso ocorra, é necessário que o instrumento utilizado tenha um som prolongado.

INSTRUMENTOS NÃO-MIXÁVEIS: alguns instrumentos, por motivos desconhecidos, não podem ser mixados (misturados) com outros mesmo em diversos canais. Esses instrumentos, geralmente os mais altos, devem ser utilizados com certa precaução sob risco de atrapalhar os outros. Abaixo, uma relação de instrumentos mixáveis e não mixáveis:

MIXÁVEIS: 00,02,03,04,05,06,07,09,12,14,16,24.

NÃO-MIXÁVEIS: 01,08,10,11,13,15,17,18,19,20,21,22,
23,24 até o 62.

PARAMETRO "Y": quando utilizado dentro de um comando PLAY, serve para alterar o som de alguns instrumentos pré-definidos. Caso esteja dentro de um canal de bateria, ele irá alterar o som dos instrumentos de bateria (como no comando CALL AUDREG).

A sintaxe de sua utilização é:Y<r>,<n>, onde <r> é um registrador de som entre 0 e 255 onde não são aceitos todos os números, apenas alguns. O <n> é um valor qualquer entre 0 e 255 atribuído a esse registrador.Ex:

PLAY#2,"@40V15Y32,98Y16,72@W32Y16,70@W32Y16,68@W32Y16,66@W32"

Com esse exemplo você tocou um BAIXO (@40) com diminuição do PITCH (Y16,...). O parâmetro "Y32,..." faz a alteração no valor do TRANSPOSE.

Note que foi utilizado o parâmetro "@W" para se prolongar o tempo de execução da nota, uma vez que o parâmetro "Y" não faz isso (maiores detalhes veja o comando CALL AUDREG).

APÊNDICE C - RELAÇÃO DE INSTRUMENTOS

A seguir temos uma tabela com todos os instrumentos do FM.

Nº:Nome:	Nº:Nome:
00 Piano I	32 Piano III
01 Piano II	33 Eletric Piano II
02 Violin	34 Santool II
03 Flute	35 Brass
04 Clarinet	36 Flute II
05 Oboe	37 Clavicod II
06 Trumpet	38 Clavicod III
07 Pipe Organ	39 Koto II
08 Xylophone	40 Pipe Organ II
09 Organ	41 PohdsPLA
10 Guitar	42 RohdsPLA
11 Santool	43 Orch L
12 Eletric Piano	44 Orch R
13 Clavicod	45 Synth Violin
14 Harp Sicked	46 Synth Organ
15 Harp Sicked II	47 Synth Brass
16 Vibraphone	48 Tube
17 Koto	49 Shamisen
18 Taiko	50 Magical
19 Engine	51 Huwawa
20 UFO	52 Wonder Flute
21 Synth Bell	53 Hard Rock
22 Chime	54 Machine
23 Synth Bass	55 Machine V
24 Synthesizer	56 Comic
25 Synth Percussion	57 SE Comic
26 Synth Rhythm	58 SE Laser
27 Harm Drum	59 SE Noise
28 Cow Bell	60 SE Star
29 Close Hi-Hat	61 SE Star II
30 Snare Drum	62 Engine II
31 Bass Drum	63 Silence

APÊNDICE D - EXEMPLOS DE MÚSICAS

Abaixo existem exemplos de duas músicas que demonstram algumas das funções do FM Music. A primeira ("UNCHAINED MELODY") utiliza 4 canais, sendo 2 concatenados (para efeito de reverberação), e 4 instrumentos: FLAUTA, PIANO, VIOLINO E BAIXO. A bateria também foi acionada.

A segunda música ("THEME FROM OVER THE NET") usa 2 canais concatenados (ou seja, 6 vozes) e 1 único instrumento, o CLAVICOD, que tem o som semelhante ao de uma guitarra de heavy metal. É uma música retirada de um jogo do micro AMIGA 500 utilizando recursos de eco. A bateria foi totalmente modificada e faz uso do PSG para auxiliar.

- UNCHAINED MELODY -

```
Ø 'CopyRight (C) 1991 - Adaptado e Mixado por
  Flavio Monaco
1 SCREENØ,,Ø:KEYOFF
2 WIDTH4Ø:COLOR3,Ø,Ø
3 LOCATE1,2:PRINT"//=================================
===\\"
4 LOCATE1,3:PRINT"||---------------------------------
---||
7 FORZ=4TO17:LOCATE2,Z:PRINT"||
                  ||":NEXT
6 LOCATE1,18:PRINT"||--------------------------------
----||"
7 LOCATE1,19:PRINT"\\================================
====//"
8 LOCATE11,5:PRINTCHR$(34)"UNCHAINED MELODY"CHR$(34
)
9 LOCATE4,7:PRINT"Trilha Sonora Original do filme:"
1Ø LOCATE16,8:PRINT"/--------------------------------
----\"
11 LOCATE16,9:PRINT"|                 GHOST
|"
12 LOCATE16,1Ø:PRINT"\-------------------------------
----/"
13 LOCATE9,12:PRINT"Versao FM-PAC/MSX por:"
14 LOCATE12,13:PRINT"/------------\"
15 LOCATE12,14:PRINT"|FLAVIO MONACO|"
```

```
16 LOCATE12,15:PRINT"\-------------/"
17 LOCATE4,16:PRINT"COPYRIGHT (C) 1991 - MC
MONACO'S"
18 _MUSIC(1,Ø,1,1,2,2)
19 POKE&HFA5C,4Ø:POKE&HFA7C,6Ø
2Ø T=1ØØ:'Tempo de andamento da musica
21 PLAY#2,"t=t;","t=t;","t=t;","t=t;","t=t;"
22 BT$="v5b!c8c8c8s!c8c8b!c8b!c8c8c8s!c8c8b!c8"
23 EB$="v5b!c8c8c8s!c8c8b!c8b!c8s!c8s!c8b!c8c8b!c8"
24 'Piano Acomp.
25 P1$="@ØØv1118o5ceg>c<geceg>c<ge"
26 P2$="@ØØv1118o4a>ceaec<a>ceaec"
27 P3$="@ØØv1118o4fa>cfc<afa>cfc<a"
28 P4$="@ØØv1118o4gb>dgd<bgb>dgd<b"
29 P5$="@ØØv1118o4egb>e<bgegb>e<bg"
3Ø P6$="@ØØv1118o4fa>cfc<agb>dgd<b"
31 P7$="@ØØv1118o4fa>cfc<aegb>e<bg"
32 P8$="@ØØv1118o4fa>cfc<agb>dgd<b"
33 P9$="@ØØv1118o5ceg>c<geceg>c<ge"
34 PF$="@ØØv1118o5c"
35 'Flute Solo
36 W1$="@36v15o5c1&c4&c8d8"
37 W2$="@36v15o5c1&c4&c8d8"
38 W3$="@36v15o5e4c1&c8<a8"
39 W4$="@36v15o4g2&g8b8>d2e8d8"
4Ø W5$="@36v15o5c1&c4&c8d8"
41 W6$="@36v15o5c1&c8<e4f8"
42 W7$="@36v15o4g1&g2"
43 W8$="@36v15o4g1g8>e4d8"
44 X1$="@36v15o5c1&c4&c8d8"
45 X2$="@36v15o5c1&c4&c8d8"
46 X3$="@36v15o5e4c1c8d8"
47 X4$="@36v15o4b2&b4>a8g8a8g8e8d8"
48 X5$="@36v15o5c1&c4&c8d8"
49 X6$="@36v15o5c1&c4&c8c8"
5Ø X7$="@36v15o4>g1&g2"
51 X8$="@36v15o4>g1g8f8e8d8"
52 Y1$="@36v15o5e4c8<g1&g8&"
53 Y2$="@36v15o4g1&g8>d8d8d8"
54 Y3$="@36v15o5d4c8<e1&e8&"
55 Y4$="@36v15o4e1&e4>c8<b8"
56 Y5$="@36v15o4a1&a2"
```

```
57 Y6$="@36v15o5d2e8d8e8g4&g8e8d8"
58 Y7$="@36v15o5c1&c2"
59 Y8$=""
60 Z1$="@36v15o4f4g4a8>c8<b2g8b8"
61 Z2$="@36v15o4a2f8a8g2&g4"
62 Z3$="@36v15o4f4g4a8>c8d2d8c8"
63 Z4$="@36v15o5e4d4c1"
64 Z5$="@36v15o4f4g4a8>c8<b2>c8<b8"
65 Z6$="@36v15o4a2f8a8g2&g4"
66 Z7$="@36v15o4f4g4a8>c8d2e8d8"
67 Z8$="@36v15o5e8d8c1&c4"
68 ZF$="@36v15o5c1"
69 'Violin Acomp.
70 V1$="@02v10o6c1&c2"
71 V2$="@02v10o5a1&a2"
72 V3$="@02v10o5f1&f2"
73 V4$="@02v10o5g1&g2"
74 V5$="@02v10o5e1&e2"
75 V6$="@02v10o5f2&f4&g2&g4"
76 V7$="@02v10o5f2&f4&e2&e4"
77 V8$="@02v10o5f2&f4&g2&g4"
78 V9$="@02v10o6c1&c2"
79 VF$="@02v08o6c1"
80 'Bass
81 B1$="@24v15o3c2&c8<g8>c4&c8<b4b8"
82 B2$="@24v15o2a2&a8e8a4&a8g4g8"
83 B3$="@24v15o2f2&f8c8f4&f8e4f8"
84 B4$="@24v15o2g2&g8d8g4&g8a4b8"
85 BE$="@24v15o2g2&g8d8g4&g8&g4d8"
86 B5$="@24v15o2e2&e8<b8>e4&e8d4e8"
87 B6$="@24v15o2f2&f8c8g2&g8d8"
88 B7$="@24v15o2f2&f8c8e2&e8<b8"
89 B8$="@24v15o2f2&f8c8g2&g8d8"
90 B9$="@24v15o3c2&c8<g8>c4&c8&c4<g8"
91 BF$="@24v15o3c1"
92 PLAY#2,P1$,W1$,"r",B1$,BT$
93 PLAY#2,P2$,W2$,"r",B2$,BT$
94 PLAY#2,P3$,W3$,"r",B3$,BT$
95 PLAY#2,P4$,W4$,"r",B4$,BT$
96 PLAY#2,P1$,W5$,"r",B1$,BT$
97 PLAY#2,P2$,W6$,"r",B2$,BT$
98 PLAY#2,P4$,W7$,"r",BE$,BT$
```

```
99 PLAY#2,P4$,W8$,"r",B4$,BT$
100 PLAY#2,P1$,X1$,V1$,B1$,BT$
101 PLAY#2,P2$,X2$,V2$,B2$,BT$
102 PLAY#2,P3$,X3$,V3$,B3$,BT$
103 PLAY#2,P4$,X4$,V4$,B4$,BT$
104 PLAY#2,P1$,X5$,V1$,B1$,BT$
105 PLAY#2,P2$,X6$,V2$,B2$,BT$
106 PLAY#2,P4$,X7$,V4$,BE$,BT$
107 PLAY#2,P4$,X8$,V4$,B4$,BT$
108 PLAY#2,P1$,Y1$,V1$,B1$,BT$
109 PLAY#2,P4$,Y2$,V4$,B4$,BT$
110 PLAY#2,P3$,Y3$,V3$,B3$,BT$
111 PLAY#2,P5$,Y4$,V5$,B5$,BT$
112 PLAY#2,P3$,Y5$,V3$,B3$,BT$
113 PLAY#2,P4$,Y6$,V4$,B4$,BT$
114 PLAY#2,P1$,Y7$,V1$,B1$,BT$
115 PLAY#2,P1$,Y8$,V1$,B1$,EB$
116 PLAY#2,P1$,W1$,V1$,B1$,BT$
117 PLAY#2,P2$,W2$,V2$,B2$,BT$
118 PLAY#2,P3$,W3$,V3$,B3$,BT$
119 PLAY#2,P4$,W4$,V4$,B4$,BT$
120 PLAY#2,P1$,W5$,V1$,B1$,BT$
121 PLAY#2,P2$,W6$,V2$,B2$,BT$
122 PLAY#2,P4$,W7$,V4$,BE$,BT$
123 PLAY#2,P4$,W8$,V4$,B4$,BT$
124 PLAY#2,P1$,X1$,V1$,B1$,BT$
125 PLAY#2,P2$,X2$,V2$,B2$,BT$
126 PLAY#2,P3$,X3$,V3$,B3$,BT$
127 PLAY#2,P4$,X4$,V4$,B4$,BT$
128 PLAY#2,P1$,X5$,V1$,B1$,BT$
129 PLAY#2,P2$,X6$,V2$,B2$,BT$
130 PLAY#2,P4$,X7$,V4$,BE$,BT$
131 PLAY#2,P4$,X8$,V4$,B4$,BT$
132 PLAY#2,P1$,Y1$,V1$,B1$,BT$
133 PLAY#2,P4$,Y2$,V4$,B4$,BT$
134 PLAY#2,P3$,Y3$,V3$,B3$,BT$
135 PLAY#2,P5$,Y4$,V5$,B5$,BT$
136 PLAY#2,P3$,Y5$,V3$,B3$,BT$
137 PLAY#2,P4$,Y6$,V4$,B4$,BT$
138 PLAY#2,P1$,Y7$,V1$,B1$,BT$
139 PLAY#2,P1$,Y8$,V1$,B1$,EB$
140 PLAY#2,P6$,Z1$,V6$,B6$,BT$
```

```
141 PLAY#2,P7$,Z2$,V7$,B7$,BT$
142 PLAY#2,P8$,Z3$,V8$,B8$,BT$
143 PLAY#2,P9$,Z4$,V9$,B9$,BT$
144 PLAY#2,P6$,Z5$,V6$,B6$,BT$
145 PLAY#2,P7$,Z6$,V7$,B7$,BT$
146 PLAY#2,P8$,Z7$,V8$,B8$,BT$
147 PLAY#2,P9$,Z8$,V9$,B9$,EB$
148 PLAY#2,P1$,W1$,V1$,B1$,BT$
149 PLAY#2,P2$,W2$,V2$,B2$,BT$
150 PLAY#2,P3$,W3$,V3$,B3$,BT$
151 PLAY#2,P4$,W4$,V4$,B4$,BT$
152 PLAY#2,P1$,W5$,V1$,B1$,BT$
153 PLAY#2,P2$,W6$,V2$,B2$,BT$
154 PLAY#2,P4$,W7$,V4$,BE$,BT$
155 PLAY#2,P4$,W8$,V4$,B4$,BT$
156 PLAY#2,P1$,X1$,V1$,B1$,BT$
157 PLAY#2,P2$,X2$,V2$,B2$,BT$
158 PLAY#2,P3$,X3$,V3$,B3$,BT$
159 PLAY#2,P4$,X4$,V4$,B4$,BT$
160 PLAY#2,P1$,X5$,V1$,B1$,BT$
161 PLAY#2,P2$,X6$,V2$,B2$,BT$
162 PLAY#2,P4$,X7$,V4$,BE$,BT$
163 PLAY#2,P4$,X8$,V4$,B4$,BT$
164 PLAY#2,P1$,Y1$,V1$,B1$,BT$
165 PLAY#2,P4$,Y2$,V4$,B4$,BT$
166 PLAY#2,P3$,Y3$,V3$,B3$,BT$
167 PLAY#2,P5$,Y4$,V5$,B5$,BT$
168 PLAY#2,P3$,Y5$,V3$,B3$,BT$
169 PLAY#2,P4$,Y6$,V4$,B4$,BT$
170 PLAY#2,P1$,Y7$,V1$,B1$,BT$
171 PLAY#2,P1$,Y8$,V1$,B1$,EB$
172 PLAY#2,P6$,Z1$,V6$,B6$,BT$
173 PLAY#2,P7$,Z2$,V7$,B7$,BT$
174 PLAY#2,P8$,Z3$,V8$,B8$,BT$
175 PLAY#2,P9$,Z4$,V9$,B9$,BT$
176 PLAY#2,P6$,Z5$,V6$,B6$,BT$
177 PLAY#2,P7$,Z6$,V7$,B7$,BT$
178 PLAY#2,P8$,Z7$,V8$,B8$,BT$
179 PLAY#2,P9$,Z8$,V9$,B9$,EB$
180 T=100
181 PLAY#2,"t=t;","t=t;","t=t;","t=t;","t=t;"
182 PLAY#2,P8$,Z8$,V8$,B8$
```

```
183 T=8Ø
184 PLAY#2,"t=t;","t=t;","t=t;","t=t;","t=t;"
185 PLAY#2,P9$,Z8$,V9$,B9$
186 PLAY#2,PF$,ZF$,VF$,BF$
187 _PLAY(Ø,Y):IF Y THEN 187 ELSE SCREENØ:END
```

- THEME FROM OVER THE NET -

```
Ø 'CopyRight (C) 1991 - Adaptado e Mixado por
Flávio Mônaco
1 _MUSIC
2 SCREENØ,,Ø:KEYOFF:WIDTH4Ø:COLOR8,1
3 WIDTH4Ø:COLOR8,Ø,Ø
4 COLOR=(8,Ø,Ø,Ø)
5 LOCATE1,2:PRINT"//===============================
===\\"
6 LOCATE1,3:PRINT"¦¦-------------------------------
---¦¦
7 FORZ=4TO17:LOCATE2,Z:PRINT"¦¦
             ¦¦":NEXT
8 LOCATE1,18:PRINT"¦¦------------------------------
----¦¦"
9 LOCATE1,19:PRINT"\\==============================
====//"
1Ø LOCATE9,6:PRINT"THEME FROM OVER THE NET"
11 LOCATE7,8:PRINT"Jogo Original do AMIGA 5ØØ"
12 LOCATE13,1Ø:PRINT"Adaptado por:"
13 LOCATE12,12:PRINT"/-------------\"
14 LOCATE12,13:PRINT"¦FLAVIO MONACO¦"
15 LOCATE12,14:PRINT"\-------------/"
16 LOCATE4,16:PRINT"COPYRIGHT (C) 1991 - MC
MONACO'S"
17 FORZ=ØTO7STEP.25
18 COLOR=(8,Z,Ø,Ø)
19 NEXTZ:COLOR=(8,7,1,1)
2Ø _MUSIC(1,Ø,2,2,2):DEFSTRA-S:POKE&HFA3C,4Ø:POKE&H
FA5C,4Ø:POKE&H7C,4Ø
21 SOUND7,247:SOUND6,1:DIMG1(32),G2(32),G3(32),B(32
),P(32)
22 V=8:VE=4:IM="v=v;@13t12Ø":IB="v15t12Ø"
23 PLAY#2,IM,IM,IM,IB,IB
24 'guitar 1
```

```
25 G1(1)="o3f#8f#8>f#8<f#8r4>f#8<f#8f#8f#8>f#8<f#8r
4>f#8<f#8
26 G1(2)="o3a8a8>a8<a8r4>a8<a8e8e8>e8<e8r4>e8<e8
27G1(3)="o3f#8f#8>f#8<f#8v=ve;f#8vØf#8v=v;>f#8<f#8f
#8f#8>f#8<f#8v=ve;f#8vØf#8v=v;>f#8<f#8
28 G1(4)="o3a8a8>a8<a8v=ve;a8vØa8v=v;>a8<a8e8e8>e8<
e8v=ve;e8vØe8v=v;>e8<e8
29G1(5)="o3f#8f#8>f#8<f#8v=ve;f#8vØf#8v=v;>f#8<f#8f
#8f#8>f#8<f#8v=ve;f#8vØf#8v=v;>f#8<f#8
3Ø G1(6)="o3a8a8>a8<a8v=ve;a8vØa8v=v;>a8<a8e8e8>e8<
e8v=ve;e8vØe8v=v;>e8<e8
31 G1(7)="o3f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
32 G1(8)="o3f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
33 G1(9)="o3f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
34 G1(1Ø)="o3a8a8>a8<a8a8a8>a8<a8e8e8>e8<e8e8e8>e8<
e8"
35 G1(11)="o3f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8
<f#8f#8f#8>f#8<f#8
36 G1(12)="o3a8a8>a8<a8a8a8>a8<a8e8e8>e8<e8e8e8>e8<
e8"
37 G1(13)="o3f#8v=ve;f#8v=v;a8f#8a8f#8a8b8v=ve;b8v=
v;b8a8f#8e8c#8e8f#8"
38G1(14)="o3f#8v=ve;f#8v=v;a8f#8a8f#8a8b8v=ve;b8v=v
;>c8v=ve;c8v=v;c#8e8c#8e8f#8
39 G1(15)="o3f#8v=ve;f#8v=v;a8f#8a8f#8a8b8v=ve;b8v=
v;b8a8f#8e8c#8e8f#8"
4ØG1(16)="o3f#8v=ve;f#8v=v;a8f#8a8f#8a8b8v=ve;b8v=v
;>c8v=ve;c8v=v;c#8e8c#8e8f#8
41 'guitar 2
42 G2(1)="o2f#8f#8>f#8<f#8r4>f#8<f#8f#8f#8>f#8<f#8
r4>f#8<f#8
43 G2(2)="o2a8a8>a8<a8r4>a8<a8e8e8>e8<e8r4>e8<e8
44G2(3)="o2f#8f#8>f#8<f#8v=ve;f#8vØf#8v=v;>f#8<f#8f
#8f#8>f#8<f#8v=ve;f#8vØf#8v=v;>f#8<f#8
45 G2(4)="o2a8a8>a8<a8v=ve;a8vØa8v=v;>a8<a8e8e8>e8<
e8v=ve;e8vØe8v=v;>e8<e8
46G2(5)="o2f#8f#8>f#8<f#8v=ve;f#8vØf#8v=v;>f#8<f#8f
#8f#8>f#8<f#8v=ve;f#8vØf#8v=v;>f#8<f#8
47 G2(6)="o2a8a8>a8<a8v=ve;a8vØa8v=v;>a8<a8e8e8>e8<
```

```
e8v=ve;e8v0e8v=v;>e8<e8
48 G2(7)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
49 G2(8)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
50 G2(9)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
51 G2(10)="o2a8a8>a8<a8a8a8>a8<a8e8e8>e8<e8e8e8>e8<
e8"
52 G2(11)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8
<f#8f#8f#8>f#8<f#8
53 G2(12)="o2a8a8>a8<a8a8a8>a8<a8e8e8>e8<e8e8e8>e8<
e8"
54 G2(13)="o2f#8v=ve;f#8v=v;a8f#8a8f#8a8b8v=ve;b8v=
v;b8a8f#8e8c#8e8f#8"
55 G2(14)="o2f#8v=ve;f#8v=v;a8f#8a8f#8a8b8v=ve;b8v=
v;>c8v=ve;c8v=v;c#8e8c#8e8f#8
56 G2(15)="o2f#8v=ve;f#8v=v;a8f#8a8f#8a8b8v=ve;b8v=
v;b8a8f#8e8c#8e8f#8"
57 G2(16)="o2f#8v=ve;f#8v=v;a8f#8a8f#8a8b8v=ve;b8v=
v;>c8v=ve;c8v=v;c#8e8c#8e8f#8
58 'guitar 3
59 G3(5)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
60 G3(6)="o2a8a8>a8<a8a8a8>a8<a8e8e8>e8<e8e8e8>e8<e
8"
61 G3(7)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
62 G3(8)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
63 G3(9)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8<
f#8f#8f#8>f#8<f#8
64 G3(10)="o2a8a8>a8<a8a8a8>a8<a8e8e8>e8<e8e8e8>e8<
e8"
65 G3(11)="o2f#8f#8>f#8<f#8f#8f#8>f#8<f#8f#8f#8>f#8
<f#8f#8f#8>f#8<f#8
66 G3(12)="o2a8a8>a8<a8a8a8>a8<a8e8e8>e8<e8e8e8>e8<
e8"
67 G3(13)="o2f#8f#8f#8f#8f#8f#8f#8f#8b8b8b8b8b8b8b8
b8
68 G3(14)="o2f#8f#8f#8f#8f#8f#8f#8f#8e8e8e8e8e8e8e8
e8
```

```
69 G3(15)="o2f#8f#8f#8f#8f#8f#8f#8f#8b8b8b8b8b8b8b8
b8
70 G3(16)="o2f#8f#8f#8f#8f#8f#8f#8f#8e8e8e8e8e8e8e8
e8
71 'snare&bass drums
72 BP(1)="Y22,ØY23,ØY24,9ØV15B4M4B8B8M4B4M4B8B8M4":
BP (2)="B8B8M8.B16B8.B16M8B8B16B16B8M4B16B16M8B8M8"
:BP(4)="Y22,ØB8B8M8.B16B8.B16M8B8B16B16B8M4Y24,255M
16M16Y24,18ØM16M16Y24,6ØM16M16Y22,5ØB16B16"
73 B(9)=BP(1)
74 B(1Ø)=BP(1)
75 B(11)=BP(1)
76 B(12)=BP(2)
77 B(13)=BP(1)
78 B(14)=BP(1)
79 B(15)=BP(1)
8Ø B(16)=BP(4)
81 B(1)=BP(1)
82 B(2)=BP(1)
83 B(3)=BP(1)
84 B(4)=BP(2)
85 B(5)=BP(1)
86 B(6)=BP(1)
87 B(7)=BP(1)
88 B(8)=BP(1)
89 'chimbau&hihat
9ØBS(1)="SØL16M12ØØC8CCM3ØØØC8M12ØØCCC8CCM3ØØØC8M12
ØØCCC8CCM3ØØØC8M12ØØCCC8CCM3ØØØC8M12ØØCC":BS(2)="M1
2ØØC8CCM3ØØØC8M12ØØCCC8CCM3ØØØC8M12ØØCCC8CCM3ØØØC8M
12ØØCCCCM3ØØØC8M12ØØCCM3ØØØC8"
91 BS(4)="M12ØØC8CCM3ØØØ
C8M12ØØCCC8CCM3ØØØC8M12ØØCCC8CCM3ØØØC8M12ØØCCM3ØØØC
CM25ØØCCm2ØØØccm7ØØcc"
92 P(9)=BS(1)
93 P(1Ø)=BS(1)
94 P(11)=BS(1)
95 P(12)=BS(2)
96 P(13)=BS(1)
97 P(14)=BS(1)
98 P(15)=BS(1)
99 P(16)=BS(4)
1ØØ P(1)=BS(1)
```

```
101 P(2)=BS(1)
102 P(3)=BS(1)
103 P(4)=BS(2)
104 P(5)=BS(1)
105 P(6)=BS(1)
106 P(7)=BS(1)
107 P(8)=BS(1)
108 'play
109 FORZ=1TO8:PLAY#2,G1(Z),G2(Z),G3(Z),"","":NEXT
110 FORZ=9TO16:PLAY#2,G1(Z),G2(Z),G3(Z),B(Z),P(Z)
:NEXT
111 FORZ=1TO8:PLAY#2,G1(Z),G2(Z),G3(Z),B(Z),P(Z):
NEXT
112 FORZ=9TO16:PLAY#2,G1(Z),G2(Z),G3(Z),B(Z),P(Z):
NEXT
113PLAY#2,"o3f#1","o2f#1","o2f#1":FORX=1TO50:NEXT:F
ORT=0TO-50STEP-1 :_TRANSPOSE(T):NEXT:FORT=-50TO0:_
TRANSPOSE(T):NEXT
114 'Exit
115 FORZ=7TO0STEP-.25
116 COLOR=(8,Z,1,1)
117 NEXTZ:FORZ=1TO1000:NEXTZ:CLS
118 COLOR12,1,1:COLOR=NEW
119 END
```

CERTIFICADO DE GARANTIA E ASSISTÊNCIA TÉCNICA

A ACVS Eletrônica Ltda garante o Cartucho MSX 2+ FM contra defeitos de funcionamento e componentes pelo período de 1 (UM) ano a contar da data de aquisição do produto de acordo com a nota fiscal de compra ou aquela constante deste certificado, desde que tenha sido preenchido por pessoa autorizada pela ACVS Eletrônica.

Esta garantia perde sua validade nos seguintes casos:

- Se não forem obedecidas rigorosamente as orientações de instalação fornecidas pela ACVS Eletrônica.
- Quando constatado pelo profissional da assistência técnica que houve uso impróprio do produto.
- Ligação em redes elétricas inadequadas.
- Danos provocados por acidentes ou agentes da natureza.
- Tentativa de conserto ou manipulação por pessoa não autorizada.
- Se este certificado tiver sinais de rasura ou adulteração.

Os complementos como etiquetas, painéis, pilhas ou acessórios que sofram desgaste com o uso não estão cobertos por esta garantia.

A responsabilidade com as despesas e riscos com o transporte de ida e volta do produto até a assistência técnica é única e exclusivamente do comprador.

Quando for necessária a instalação interna de equipamentos, os gastos e riscos com o transporte do computador até a assistência técnica e seu retorno são de responsabilidade do comprador.

A ACVS Eletrônica se compromete a dar assistência técnica (não gratuita) ao restante da parte elétrica/eletrônica do computador quando nele for instalado algum equipamento de sua fabricação.

Nos casos de assistência técnica não cobertos pela garantia será cobrada uma taxa para se fazer o orçamento, e, no caso de serem efetuados os reparos, as peças trocadas e a mão-de-obra gasta no serviço também serão cobradas.

Os reparos feitos pela assistência técnica não gratuita têm garantia de 2 (dois) meses. E cobrem exclusivamente a mão-de-obra. As peças trocadas não são garantidas contra defeitos de fabricação.

Para certificar-se da validade desta garantia preencha a ficha abaixo no momento da compra.

Data de aquisição:___/___/19__.   Nº ____________________

Cartucho MSX 2+ FM para computador:
( ) Expert1.1/1.Ø   ( ) Exp.Plus
( ) Exp.DDPlus   ( ) Hot Bit

Nome legível e assinatura da pessoa autorizada e carimbo da empresa.

Nome: ____________________   Ass.: ____________________



Importação e Exportação Ltda.

Av. Paulista, 2001 - conjunto 912 - (011) 289-7694

CEP. 01311-931 - SÃO PAULO - SP